باسم اللـه الرحمان الرحيم

تقديم

أخوكم إبراهيم عدو ، موظف ببلديـة أكليم بركان المغرب ، لست مبرمجا بل متعلما من الدروس والكتب التي تعرض على مواقع الأنترنت ، تمنيت لو أن لي أستاذا يأخذ بيدي إلى عالم الإحتراف في لغة الفيجوال بيسك دوت نت ، أحببت أن أقدم كتابي هذا خدمة للإخوة والأحبة في الوطن العربي والإسلامي ، للمهتمين بلغة البرمجة الفيجوال بيسك دوت نت بالخصوص، هو كتاب ألفته بجهدي المتواضع بصيغة PDF ،عبارة عن ترجمة كتابية لما تحمله المحاضرات السمعية البصرية لأخينا الفاضل المبرمج الدكتور عاطف العراقي جزاه الله عنا خير الجزاء .

ربما يسأل سائل وما الجدوى من هذا الكتاب إذا كانت هناك أصلا فديوهات لمبرمج كبير مثل د.عاطف العراقي؟

الجواب : هو أن الذي يريد تعلم البرمجة وتطبيق ما يوجد في هذه الفيديوهات ، لا يستطيع عمل نسخ ولصق لما تحمله من أكواد إلى برنامجه ليتدرب عليها بسهولة . علاوة على ذالك ، الأكواد البرمجية على الفيديوهات التعليمية غير واضحة بالنسبة للمبتدئين عكس الدروس المكتوبة التي تحملها أوراق الكتب .

فما عليك أخي سوى تثبيت برنامج الفيجوال ستوديو 2008 أو 2010 ، و تحميل فيديوهات المبرمج الدكتور عاطف العراقي على حاسوبك ، لتبدأ التعلم .

هذا الكتاب مجاني ، ما أرجوه منكم ، هو أن تدعو لي ولعائلتي بصالح الأعمال والتوفيق .

إيميلي هو كالتالي : brahim.addou@gmail.com

اسم الكتاب : الفيجوال بيسك دوت نت : ترجمة كتابية لمحاضرات د. عاطف العراقي الجزء الأول + الجزء الثاني والأخير



المحاضرة رقم 01

أهداف تعلم البرمجة وتطبيقاتها

1 - التعرف على أساسيات البرمجة .
2 - المقدرة على صياغة المشكلات في صورة يمكن برمجتها.
3 - المقدرة على حل المشكلات باستخدام البرمجة.
4 - المقدرة على مقارنة الحلول المختلفة للمشكلات بهدف الوصول إلى أفضل هذه الحلول وأنسبها .
5 - المقدرة على تتبع الأخطاء في البرامج التي يتم تنفيذها.
6 - المقدرة على فهم البرامج المعدة سابقا.
7 - المقدرة على تطوير البرامج المعدة سابقا.

==

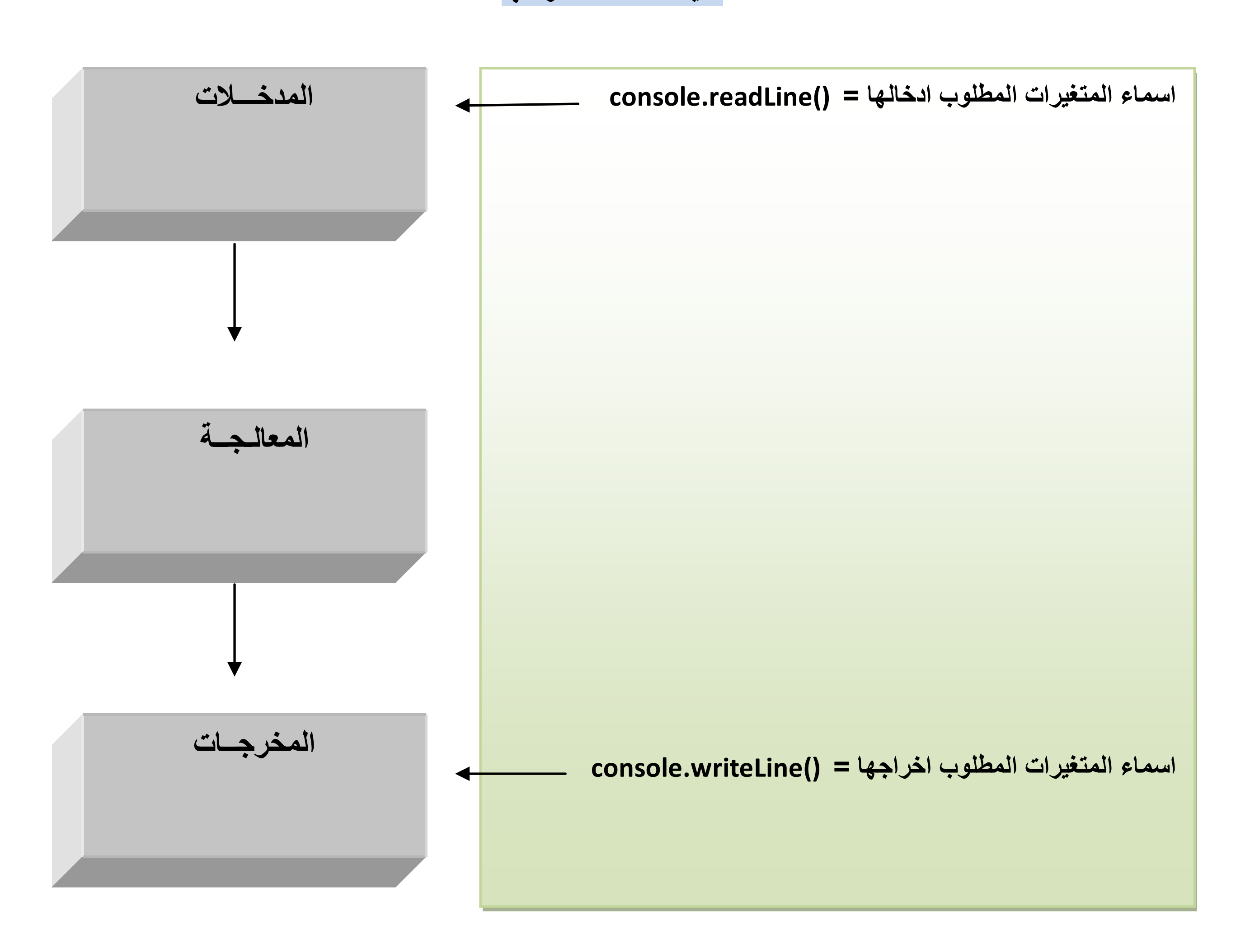
أنواع التطبيقات : Applications Types

CONSOLES APPLICATIONS

WINDOWS APPLICATIONS

WEB APPLICATIONS

صياغة المشكلات وحلها



المثال الأول :

اكتب برنامجا لحساب مجموع رقمين صحيحين

اولا: صياغة المشكلة وحلها

1-ادخال الرقمين(المدخلات)

2-جمع الرقمين(المعالجة)

3-طبع ناتج الرقمين(المخرجات)

```
Module example1
  Sub Main()
      Dim num1 As Integer
      Dim num2 As Integer
      Dim numsum As Integer
      num1 = Console.ReadLine()
      num2 = Console.ReadLine()
      numsum = num1 + num2
      Console.WriteLine(numsum)
      Console.ReadLine()
  End Sub
End Module
```

متغير صحيح

توصيف المتغيرات

ادخال الرقمين (1-عملية الإدخال)

حساب مجموع الرقمين (2-عملية المعالجة)

طباعة ناتج الجمع (3 -عملية الإخراج)

جملة ادخال لتوقيف البرنامج :

بعبارة أصح هي جملة برمجية تنتظر من المستخدم أن يدخل قيمة ما ليقرأها ، وكلمة **Line** تعني الرجوع الى سطر فارغ .

هذا البرنامج يشتغل ، الا أن فيه بعض العيوب ، نتفاداها في البرنامج الآتي، أي نحسنه ونطوره...

MEMOIRE RAM يخزن المتغيرات في الذاكرة

Num1 :	Num2 :	Numsum :	

المثا ل الاول 2/2:



```
Module example1

 Sub Main()

      Dim num1, num2, numsum As Integer

      Console.WriteLine("Entrez le premier nombre? ")

      num1 = Console.ReadLine()

      Console.WriteLine("Entrez le deuzieme nombre? ")

      num2 = Console.ReadLine()

      numsum = num1 + num2

      Console.WriteLine("la somme = " & numsum)

      Console.readLine()

End Sub

End Module
```

توصيف أو الإعلان عن المتغيرات

رسالة تطلب من المستخدم ادخا ل قيمة للمتغير num1 (الرقم الأول)

جملة الإدخال تنتظر من المستخدم إدخال القيمة للمتغير كي يخزنه.

رسالة تطلب من المستخدم ادخا ل قيمة للمتغير num2 (الرقم الثاني)

جملة الإدخال تنتظر من المستخدم إدخال القيمة للمتغير كي يخزنه

حساب مجموع الرقمين

طباعة ناتج الجمع

جملة للرجوع الى السطر تنتظرادخال قيمة كالضغط على زر لانهاء البرنامج ،

```
Module example1

Sub Main()

      Dim num1, num2, numsum As Integer

      Console.WriteLine("Entrez le premier nombre? ")

      num1 = Console.ReadLine()

      Console.WriteLine("Entrez le deuzieme nombre? ")

      num2 = Console.ReadLine()

      numsum = num1 + num2

      Console.WriteLine(num1 & "+" & num2 & "=" & numsum)

      Console.ReadLine()

  End Sub

End Module
```

نفس البرنامج السابق مع تغيير بسيط في هذه الجملة

الكلمة أو الجملة التي تقع بين مزدوجتين (" ") هي نص للطباعة ، يكتبها البرنامج كما هي ، لا تحتوي أو تحمل قيمة برمجية معادلاتية .

هذا الرمز & معناه بالإنجليزية AND إضافة نص الى نص أو حرف إلى حرف ..عطف شيء الى شيء آخر.

```
Module module1

  Sub Main()
    Dim name As String
    Dim age As Integer
    Console.WriteLine("input name?")
    name = Console.ReadLine()
    Console.WriteLine("input age?")
    age = Console.ReadLine()
    Console.WriteLine("your name is " & name & ".")
    Console.WriteLine("your age is " & age & ".")
    Console.ReadLine()
 End Sub
End Module
```

توصيف المتغيرات

متغير نصي

متغير رقمي صحيح

طباعة رسالة للمستخدم تطلب منه ادخال اسم

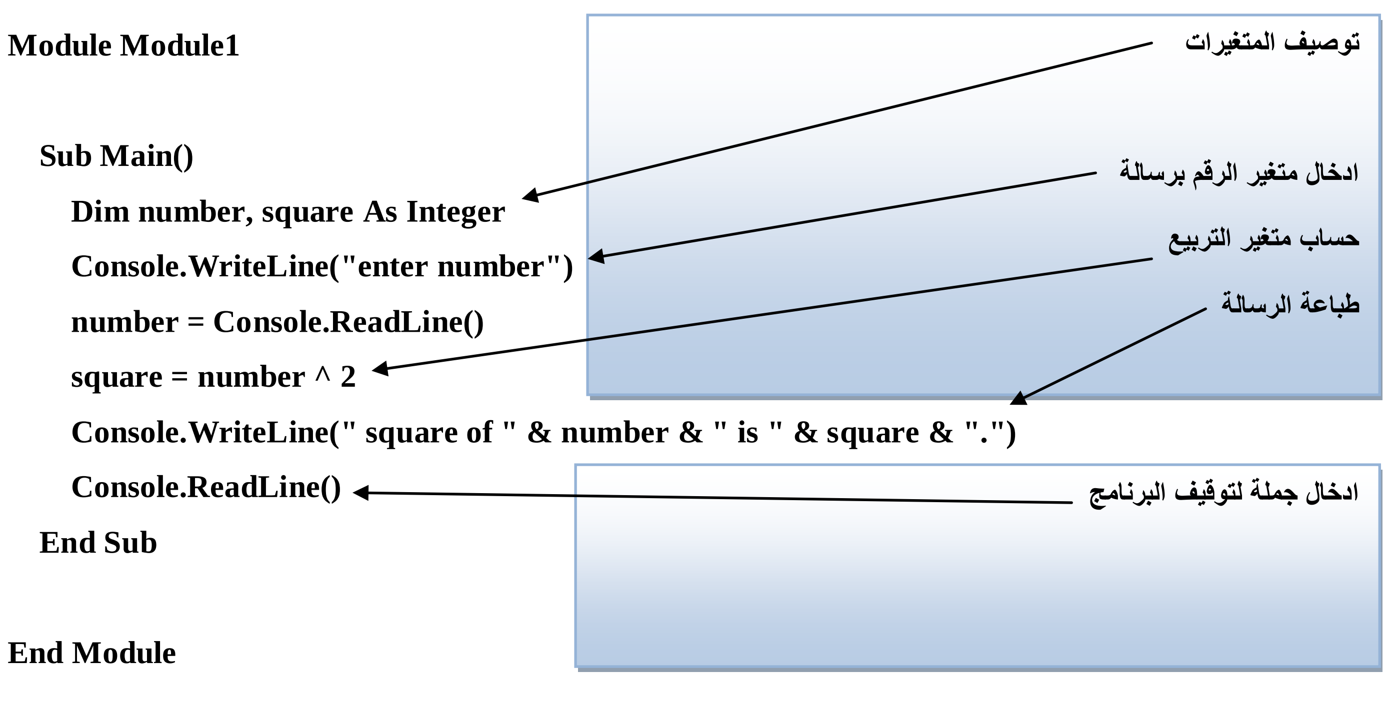
طباعة رسالة للمستخدم تطلب منه ادخال العمر

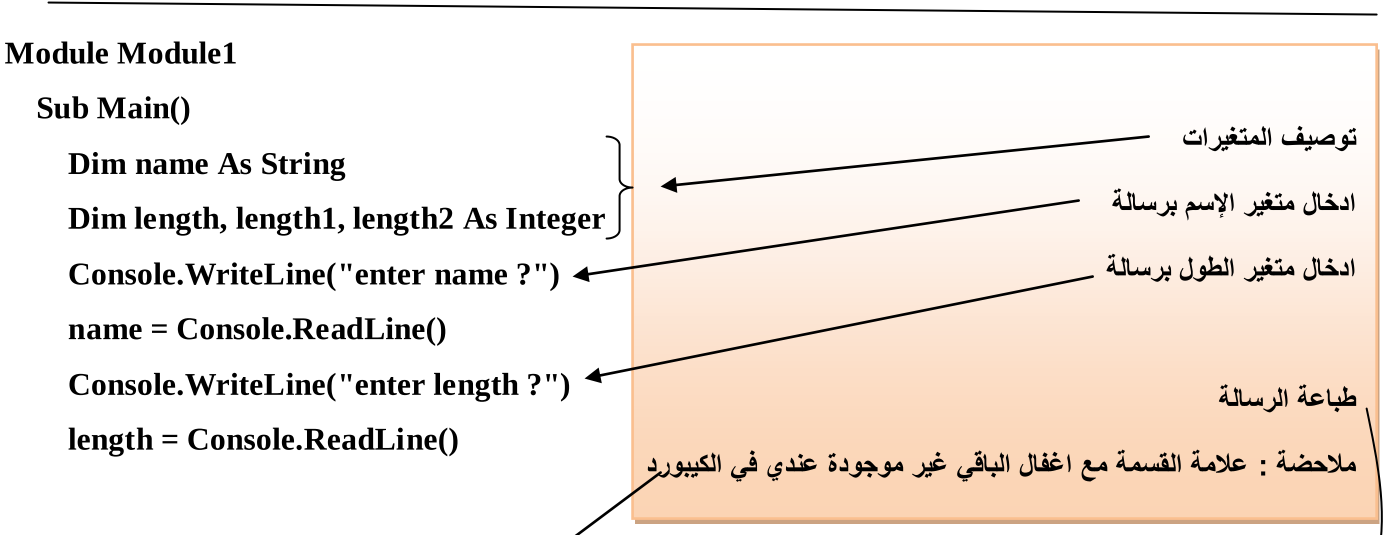
طباعة الرسائل

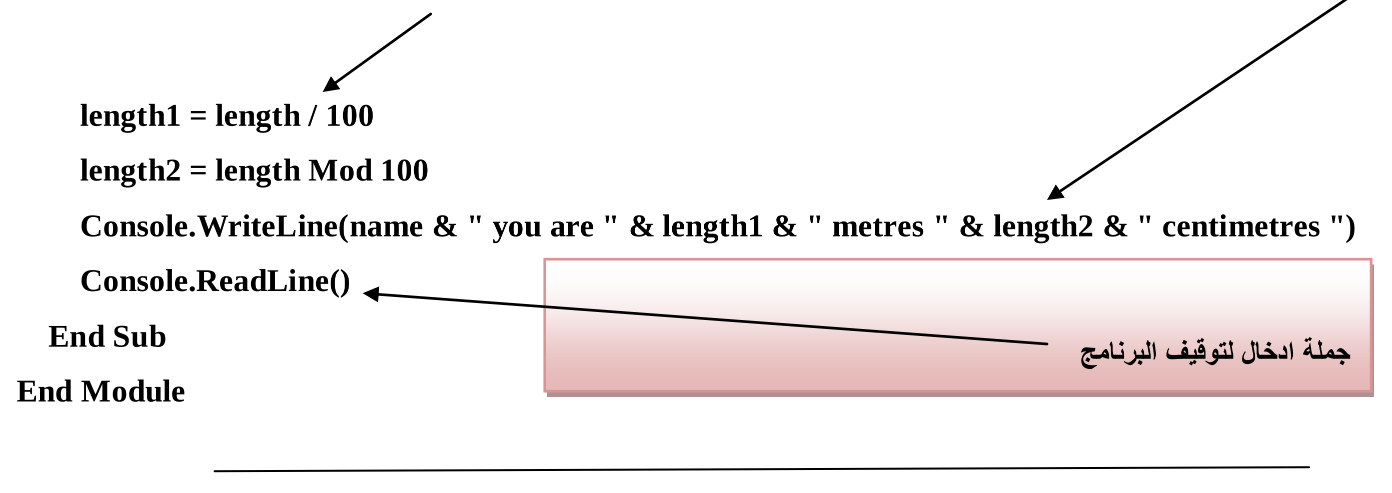
جملة ادخال لتوقيف البرنامج

ونستطيع أن نستبدلها بجملة أخرى حتى يفهمها المتلقي وهي: Console.Readkey()

يعني اضغط على أي زر من أزرار لوحة المفاتيح لانهاء البرنامج.








```
        length1 = length / 100
        length2 = length Mod 100
        Console.WriteLine(name & " you are " & length1 & " metres " & length2 & " centimetres ")
        Console.ReadLine()
    End Sub
End Module
```




```
Module Module1
    Sub Main()
        Dim name As String
        Dim length As Integer
        Console.WriteLine("enter name ?")
        name = Console.ReadLine()
        Console.WriteLine("enter length ?")
        length = Console.ReadLine()
        Console.WriteLine(name & " you are " & length / 100 & " metres and " & length Mod 100 & "
centimetres ")
        Console.ReadLine()
    End Sub
End Module
```

قم بعمل برنامج لحل المعادلة التربيعية

```
Module Module1
Sub Main()
        Dim a, b, c, x1, x2 As Integer
        Console.WriteLine("enter parameter a")
        a = Console.ReadLine()
        Console.WriteLine("enter parameter b")
        b = Console.ReadLine()
        Console.WriteLine("enter parameter c")
        c = Console.ReadLine()
        x1 = (-b + (b ^ 2 - 4 * a * c )^ 0.5 / (2 * a))
        x2 = (-b - (b ^ 2 - 4 * a * c) ^ 0.5 / (2 * a))
        Console.WriteLine(" x1 = " & x1 & " x2 = " & x2)
        Console.ReadLine()
```

توصيف المتغيرات

ادخال معاملات المعادلة برسالة

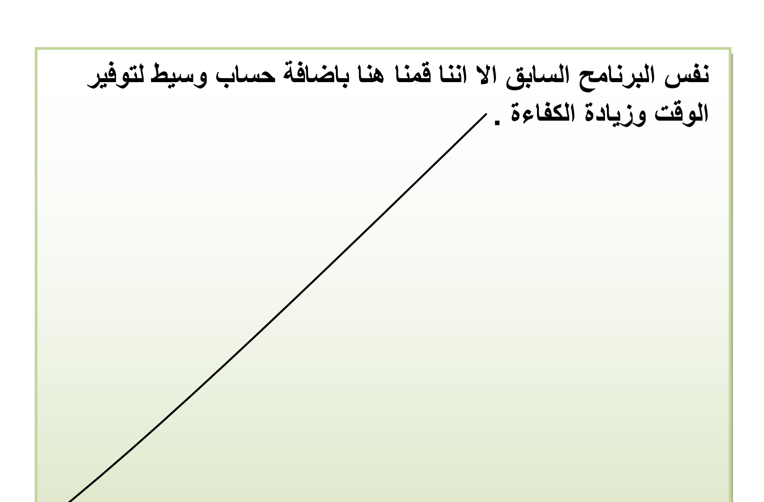
حساب قيمة الجذور

طباعة قيمة الجذور

جملة ادخال لتوقيف البرنامج

ملاحظة : يقع خطأ في البرنامج الحل في الجمل الشرطية كما سياتي

```
   End Sub
End Module
Module Module1
Sub Main()
      Dim a, b, c, d, x1, x2 As Integer
      Console.WriteLine("enter parameter a")
      a = Console.ReadLine()
      Console.WriteLine("enter parameter b")
      b = Console.ReadLine()
      Console.WriteLine("enter parameter c")
      c = Console.ReadLine()
      d = b ^ 2 - 4 * a * c
      x1 = (-b + d^ 0.5) / (2 * a)
      x2 = (-b - d^ 0.5) / (2 * a)
      Console.WriteLine(" x1 = " & x1 & " x2 = " & x2)
      Console.ReadLine()
   End Sub
End Module
```



يلزم أخذ الإحتياطات اللازمة لمنع الوقوع في أخطاء أثناء البرنامج ، لهذا سوف نستعين بالجمل الشرطية

الجمل الشرطية

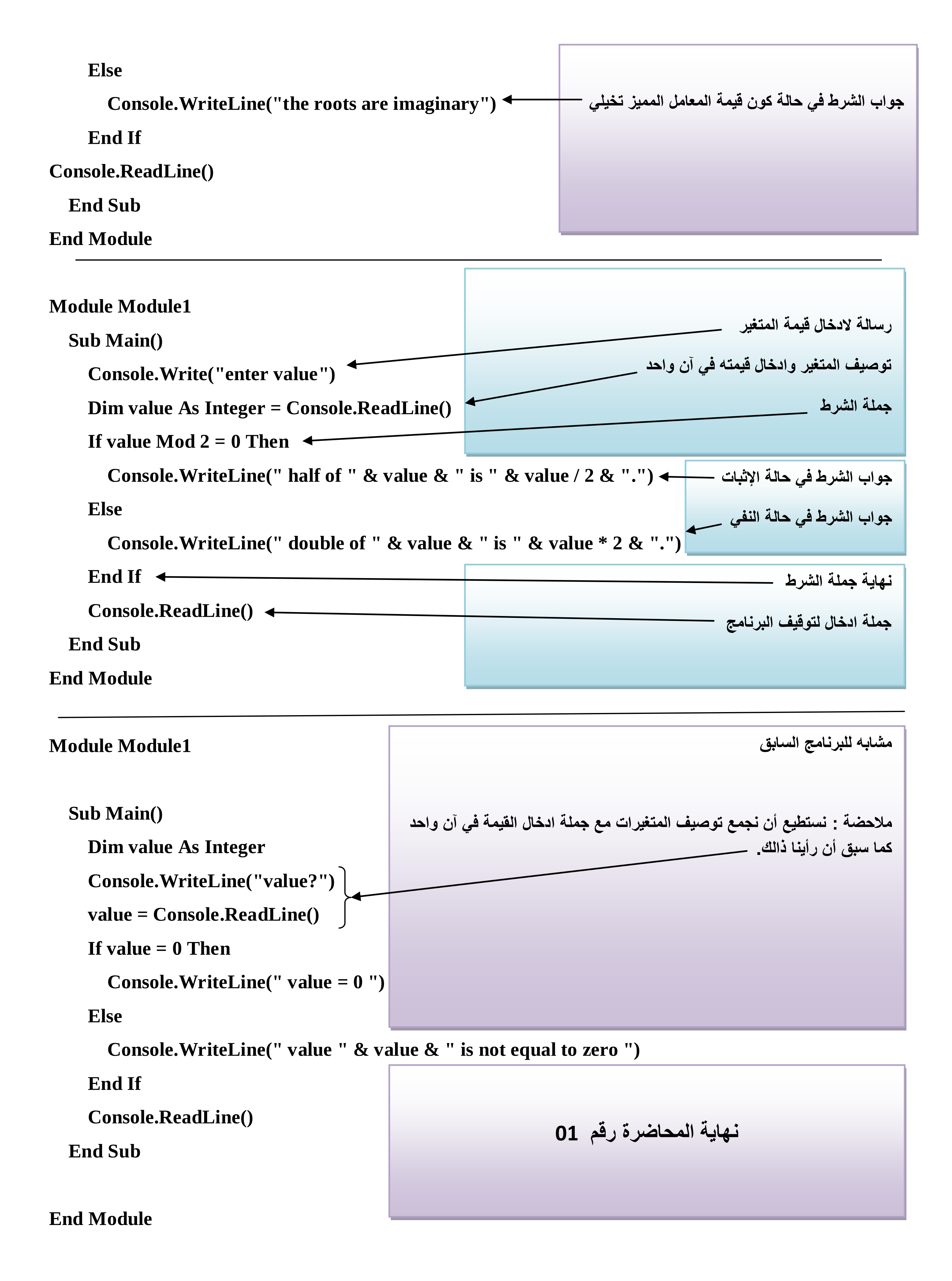
```
Module Module1
Sub Main()
      Dim a, b, c, d, x1, x2 As Decimal
      Console.WriteLine("enter parameter a ?")
      a = Console.ReadLine()
      Console.WriteLine("enter parameter b ?")
      b = Console.ReadLine()
      Console.WriteLine("enter parameter c ?")
      c = Console.ReadLine()
      d = b ^ 2 - 4 * a * c
      If d >= 0 Then
         x1 = (-b + d ^ 0.5) / (2 * a)
         x2 = (-b - d ^ 0.5) / (2 * a)
         Console.WriteLine(" x1 = " & x1 & " x2 = " & x2)
```

توصيف المتغيرات

حساب قيمة المعامل المميز

شرط حسب قيمة المعامل المميز

جواب الشرط في حالة كون قيمة المعامل المميز حقيقي




```
    Else
      Console.WriteLine("the roots are imaginary")
    End If
Console.ReadLine()
  End Sub
End Module
```

```
Module Module1
  Sub Main()
    Console.Write("enter value")
    Dim value As Integer = Console.ReadLine()
    If value Mod 2 = 0 Then
      Console.WriteLine(" half of " & value & " is " & value / 2 & ".")
    Else
      Console.WriteLine(" double of " & value & " is " & value * 2 & ".")
    End If
    Console.ReadLine()
  End Sub
End Module
```

```
Module Module1

  Sub Main()
    Dim value As Integer
    Console.WriteLine("value?")
    value = Console.ReadLine()
    If value = 0 Then
      Console.WriteLine(" value = 0 ")
    Else
      Console.WriteLine(" value " & value & " is not equal to zero ")
    End If
    Console.ReadLine()
  End Sub

End Module
```

المحاضرة رقم 02

الجمل التكرارية

تعالوا لنكتب هذا البرنامج بطريقة أخرى في البرنامج الآتي

```
Module Module1
    Sub Main()
        Console.WriteLine(1)
        Console.WriteLine(2)
        Console.WriteLine(3)
        Console.WriteLine(4)
        Console.ReadLine()
    End Sub
End Module
```

سوف يطبع لنا البرنامج رقم : 1

2

3

4

هذا البرنامج سوف يكتب لنا رقم 1 ، 2 ، 3 ثم 4 . لكن لو افترضنا أننا نريد أن يكتب لنا البرنامج من 1 الى 100 ، سوف يصعب علينا كتابة الجملة ل100 مرة . ولهذا يوجد في الفيجوال بيسك ما يسمى بالجملة التكرارية . هي حل لهذه المشكلة. تابع في البرنامج الآتي .

```
Module Module1
    Sub Main()
        Dim value As Integer
        value = value + 1
        Console.WriteLine(value)
        value = value + 1
        Console.WriteLine(value)
        value = value + 1
        Console.WriteLine(value)
        value = value + 1
        Console.WriteLine(value)
        Console.ReadLine()
    End Sub
End Module
```

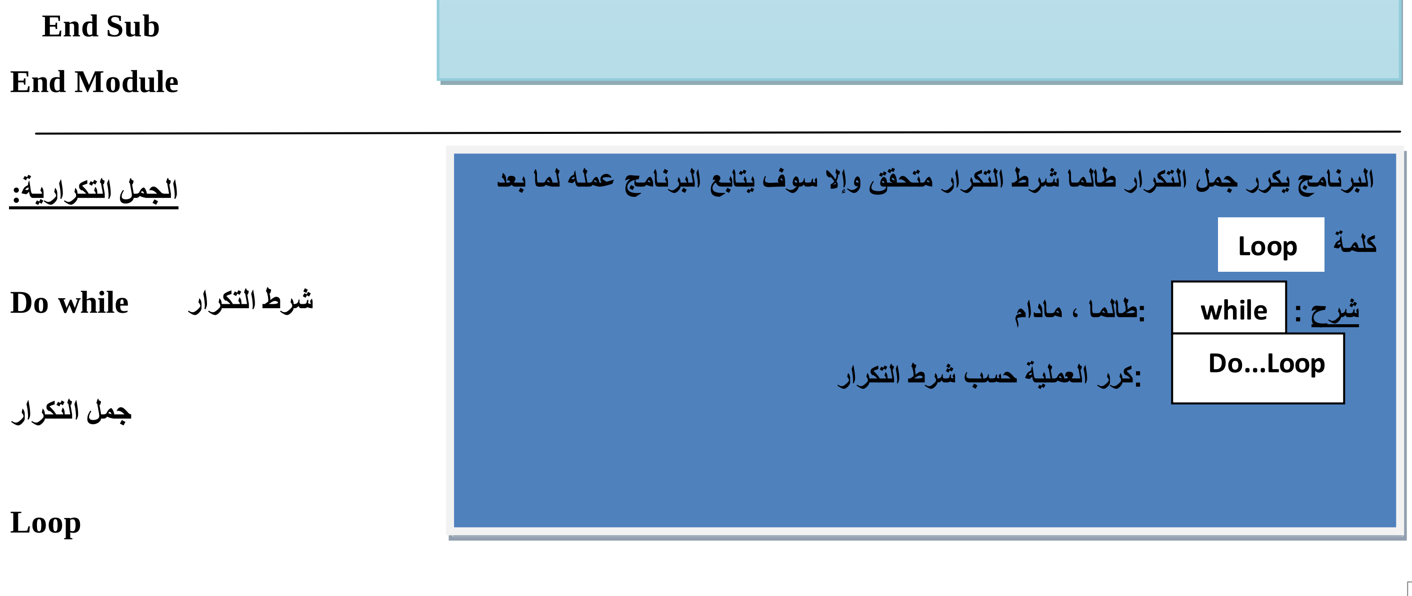
المتغير أخذ قيمة أولية التي تساوي صفر(0)

سوف يطبع لنا البرنامج رقم : 1

2

3

4



```
Module Module1

Sub Main()
    Dim value As Integer
    Do While value < 10
        value = value + 1
        Console.WriteLine(value)
    Loop
    Console.ReadLine()
  End Sub

End Module
```

لنفترض أننا نريد من البرنامج أن يطبع أو يكتب لنا الأرقام من 1 إلى 10

كما قلنا سابقا أن المتغير الرقمي دائما قيمته الأولية تساوي 0 ، ولهذا أضفنا له واحد (1)

لنفسر كيف يعمل البرنامج بالحلقة التكرارية: يمر البرنامج وينفذ سطرا سطرا ، يبدأ الأول بطباعة قيمة value التي تساوي رقم 1

يقرأ أولا جملة المعادلة ويقوم بالعملية الحسابية (المعالجة)

ثم ينزل الى السطر الموالي الذي يعني طباعة قيمة المتغير (الإخراج)

عندما يصل الى كلمة Loop يرجع الى الجملة الشرطية while ليتحقق من الشرط : هل لا زال المتغير أصغر من 10 ، ويقوم بنفس العمل الذي قام به في الأول (صعودا وهبوطا) حتى ينتهي من طباعة جميع الأرقام حتى 10 ، حين يصبح قيمة المتغير أكبر أو يساوي 10 سوف يمر البرنامج الى ما بعد Loop لينفذ ماتبقى من البرنامج أو الكود .

```
Module Module1

  Sub Main()
    Dim startvalue As Integer = 10
    Dim endvalue As Integer = 15
    Dim value As Integer = startvalue - 1
    Do While value < endvalue
        value = value + 1
        Console.WriteLine(value)
    Loop
    Console.ReadLine()

  End Sub

End Module
```

برنامج لطباعة الأعداد من 10 إلى 15

كما سبق في الدروس الماضية ، ممكن ادماج جملة التوصيف مع جملة الإدخال

جمل الإدخال هنا هو 10 و 15.

لماذا كتبنا هنا 1- ؟ لأننا نريد من البرنامج أن يبدأ طباعة الأرقام من 10 ، فإذا حذفنا 1- سوف يبدأ الطباعة من 11.

سوف يكرر 5 مرات الطباعة هبوطا وصعودا ابتداءا من Do while حتى Loop

10

11

12

13

14

15

```
Module Module1
  Sub Main()
    Dim startvalue As Integer = 10
    Dim endvalue As Integer = 15
    Dim value As Integer = startvalue
    Do While value <= endvalue
```

يمكن كتابة البرنامج السابق بهذه الصورة

لاحظ اننا أزلنا 1- وغيرنا في جملة المقارنة (أصغر أو يساوي)

غيرنا الأماكن : نكتب جملة الطباعة قبل جملة الإضافة لكي يطبع رقم 10 ثم يمر الى اضافة 1 ويصبح value=11 .

الفارق هو أن البرنامج الحالي يطبع الvalue ثم بعدها يضيف العدد 1 .عندما يصل العدد الى 16، يرجع الى جملة الشرط ليتحقق به، هل16 أصغر أو يساوي من 15 ، يتوقف البرنامج من الطباعة لأن الشرط لم يتحقق. ويمر الى مابعد كلمة LOOP ليستمر في عمله.

```
        Console.WriteLine(value)
        value = value + 1
    Loop
    Console.ReadLine()
  End Sub
End Module
```

```
Module Module1

  Sub Main()
    Console.WriteLine("enter start value ?")
    Dim startvalue As Integer = Console.ReadLine()
    Console.WriteLine("enter end value ?")
    Dim endvalue As Integer = Console.ReadLine()
    Dim value As Integer = startvalue - 1
    Do While value < endvalue
        value = value + 1
        Console.WriteLine(value)
    Loop
    Console.ReadLine()
End Sub
End Module
```

يمكن تعميم البرنامج السابق

معناه اننا نترك الإختيار للمستخدم يدخل القيم لstart value و end value .

```
Module Module1

 Sub Main()
    Dim value As Integer = 21
    Do While value > 10
        value = value - 1
        Console.WriteLine(value)
    Loop
    Console.ReadLine()
  End Sub

End Module
```

طباعة من 20 الى 10 بترتيب تنازلي

لماذا كتبنا 21 ولم نكتب 20 ، لأن البرنامج سيمر على جملة value-1 الذي يساوي 20 ويطبعه بداية .

لاحظوا أن جملة الشرط value >10 سوف يمر عليه ال value ب 11 وينزل الى الجملة التي تليها ينقص منه 1 وينزل الى الجملة التي تليها أي جملة الطباعة ويطبع النتيجة10 في آخر دوراته وينتهي . الطباعة كالآتي :

20

19

18

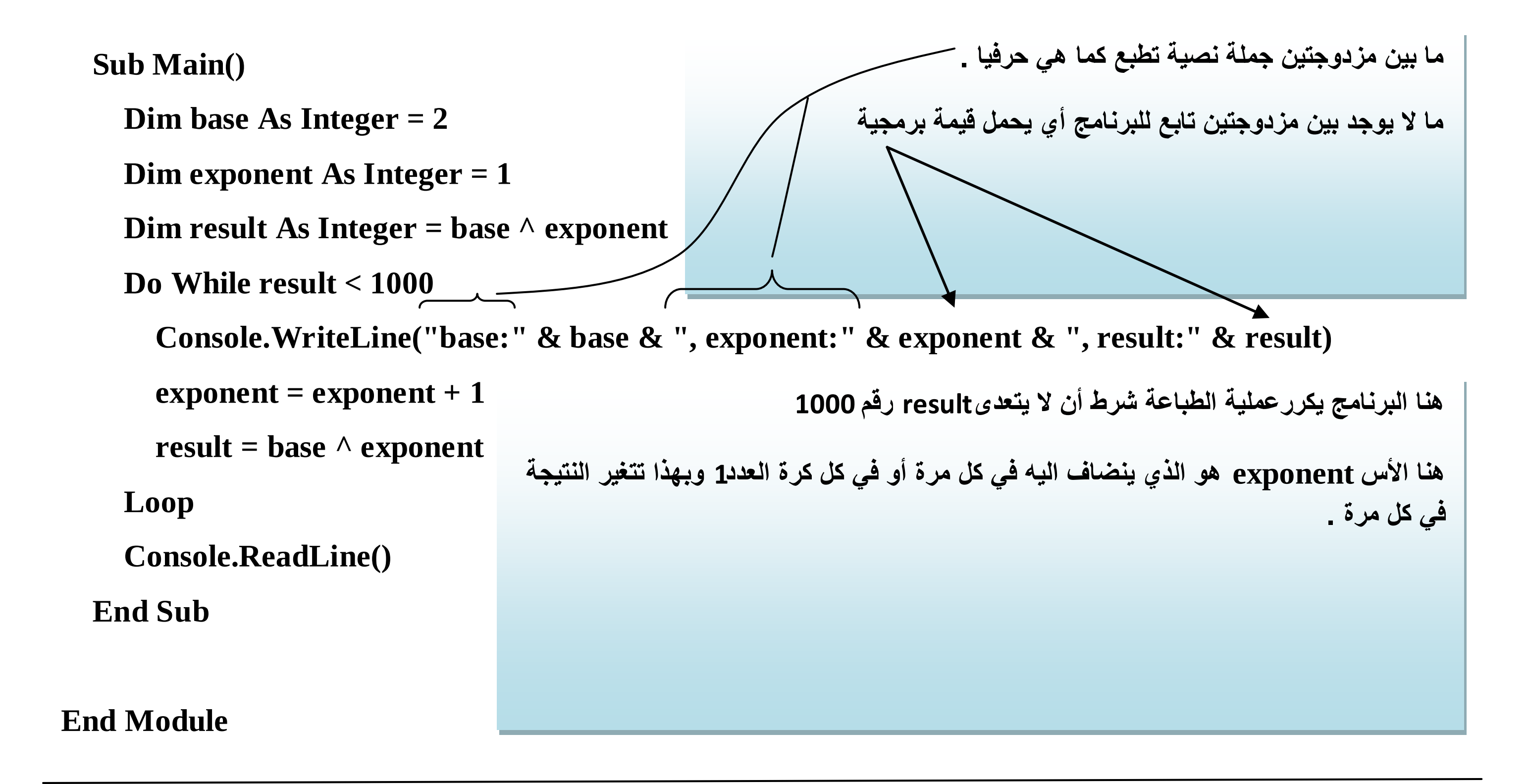
حتى 10.

هذا البرنامج يطبع بالعدد 2 الى مايقرب ال1000 بطريقة الأس

```
Module Module1

  Sub Main()
    Dim base As Integer = 2
    Dim exponent As Integer = 1
    Dim result As Integer = base ^ exponent
    Do While result < 1000
      Console.WriteLine("base:" & base & ", exponent:" & exponent & ", result:" & result)
      exponent = exponent + 1
      result = base ^ exponent
    Loop
    Console.ReadLine()
  End Sub

End Module
```



البنيـــات المتداخلـــة NESTED STRUCTURES

```
Module Module1

  Sub Main()
    Console.WriteLine("value?")
    Dim value As Integer = Console.ReadLine()
    If value = 0 Then
      Console.WriteLine("zero")
    Else
      If value > 0 Then
        Console.WriteLine("above zero")
      Else
        Console.WriteLine("bellow zero")
      End If
    End If
    Console.ReadLine()
  End Sub

End Module
```

جملة لادخال القيمة value من المستخدم

جملة توصيف ال value الذي يساوي ما سيدخله المستخدم من قيمة

جملة الشرط الذي يجب ان يتحقق

(وإلا) معناه إذا لم يتحقق الشرط السابق ذكره

شرط ثاني اذا لم يتحقق الشرط الأول

(وإلا)

معنى البرنامج أن ال value اذا كان يساوي 0 اطبع لي 0 والا (سوف نضع شرطين لان العدد اما موجب أو سالب) اذا كان الvalue أكبر من 0 اطبع لي الجملة بين قوسين ، والا اذا كان لا هذا ولا ذاك اطبع لي النص بين قوسين .

جدول الضرب

```
Module Module1

    Sub Main()
        Dim basevalue As Integer = 1
        Dim factor As Integer = 1
        Dim multiplication As Integer
        Do While basevalue < 10
            Do While factor < 10
                multiplication = basevalue * factor
                Console.WriteLine(basevalue & "*" & factor & "=" & multiplication)
                factor = factor + 1
            Loop
            basevalue = basevalue + 1
        Loop
        Console.ReadLine()
    End Sub

End Module
```

لاحظ أن هناك شرطين متداخلين أحدهما في الآخر

وأن ال LOOP الأول ينتمي للشرط الثاني والLOOP الثاني ينتمي للشرط الأول

لاحظ شكل الفقرات يدل على ذالك.

جملة الطباعة بدونها لايطبع البرنامج المعادلة التي نريدها

انتبه البرنامج سوف يقرأ الشرط الأول ثم ينزل ليقرأ الشرط الثاني حتى يصل الى الLOOP الذي ينتمي له أي ال LOOP الأول ولن يتخطاه يبدا في الصعود والنزول مايقرب9 أو 10 مرات طبعا مع تنفيذ محتوى الشرط أي الطباعة .

أخي الكريم اذا أردت أن تشاهد عمل البرنامج كيف يعمل خطوة خطوة ، فما عليك الا الضغط على الزر المتواجد أعلاه (سهم مقوس الى الأسفل تجاه أسطر صغيرة)

وكأنه معلم يشرح لك كيف يعمل البرنامج خطوة خطوة.

تعترضنا مشكلة يأخي وهي أن الfactor عندما يصل الى قيمة 10 لن يستمر في طباعة كل جدول الضرب، الحل هو أن نضيف جملة1 = factor تحت جملة الشرط الأولى، تابع الحل في البرنامج الآتي .

```
Module Module1

    Sub Main()
        Dim basevalue As Integer = 1
        Dim factor As Integer = 1
        Dim multiplication As Integer
        Do While basevalue < 10
            factor = 1
            Do While factor < 10
                multiplication = basevalue * factor
                Console.WriteLine(basevalue & "*" & factor & "=" & multiplication)
                factor = factor + 1
            Loop
            basevalue = basevalue + 1
        Loop
```

الجملة التي أضفناها لحل المشكلة ليبدأ الfactor من جديد بقيمة مبدئية 1 ويضيف عليه في كل دورة داخل الشرط الثاني

هذه الجملة تابعة للشرط الأول معناه كل ما ينتهي الشرط الثاني من عمله أي عندما يصل ال factor الى العدد 9 يخرج بعد الLOOP ليضيف على base value العدد 1 حتى يصل الى العدد 9 .

```
        Console.ReadLine()
    End Sub

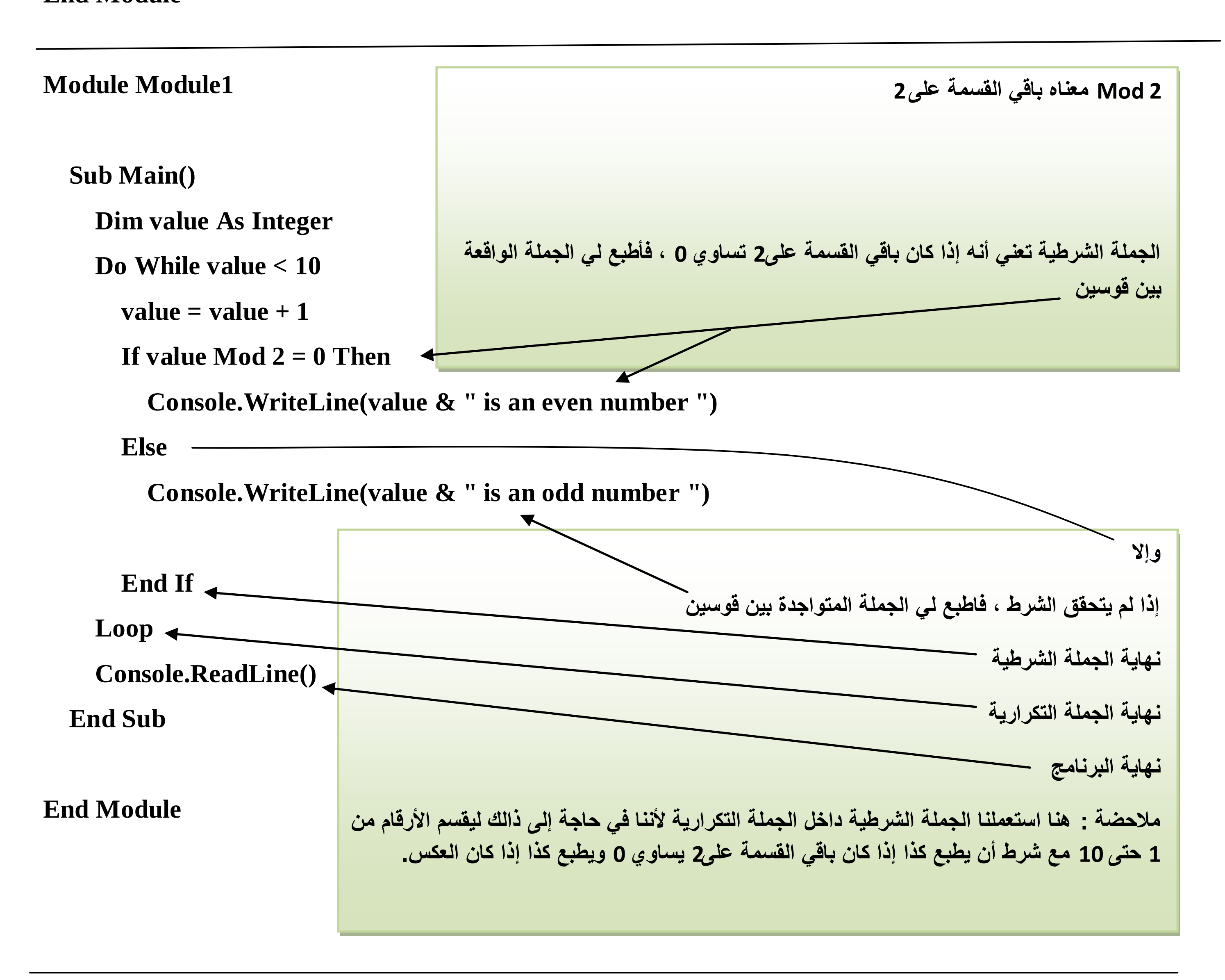
End Module
```

```
Module Module1

    Sub Main()
        Dim value As Integer
        Do While value < 10
            value = value + 1
            If value Mod 2 = 0 Then
                Console.WriteLine(value & " is an even number ")
            Else
                Console.WriteLine(value & " is an odd number ")

            End If
        Loop
        Console.ReadLine()
    End Sub

End Module
```



```
Module Module1

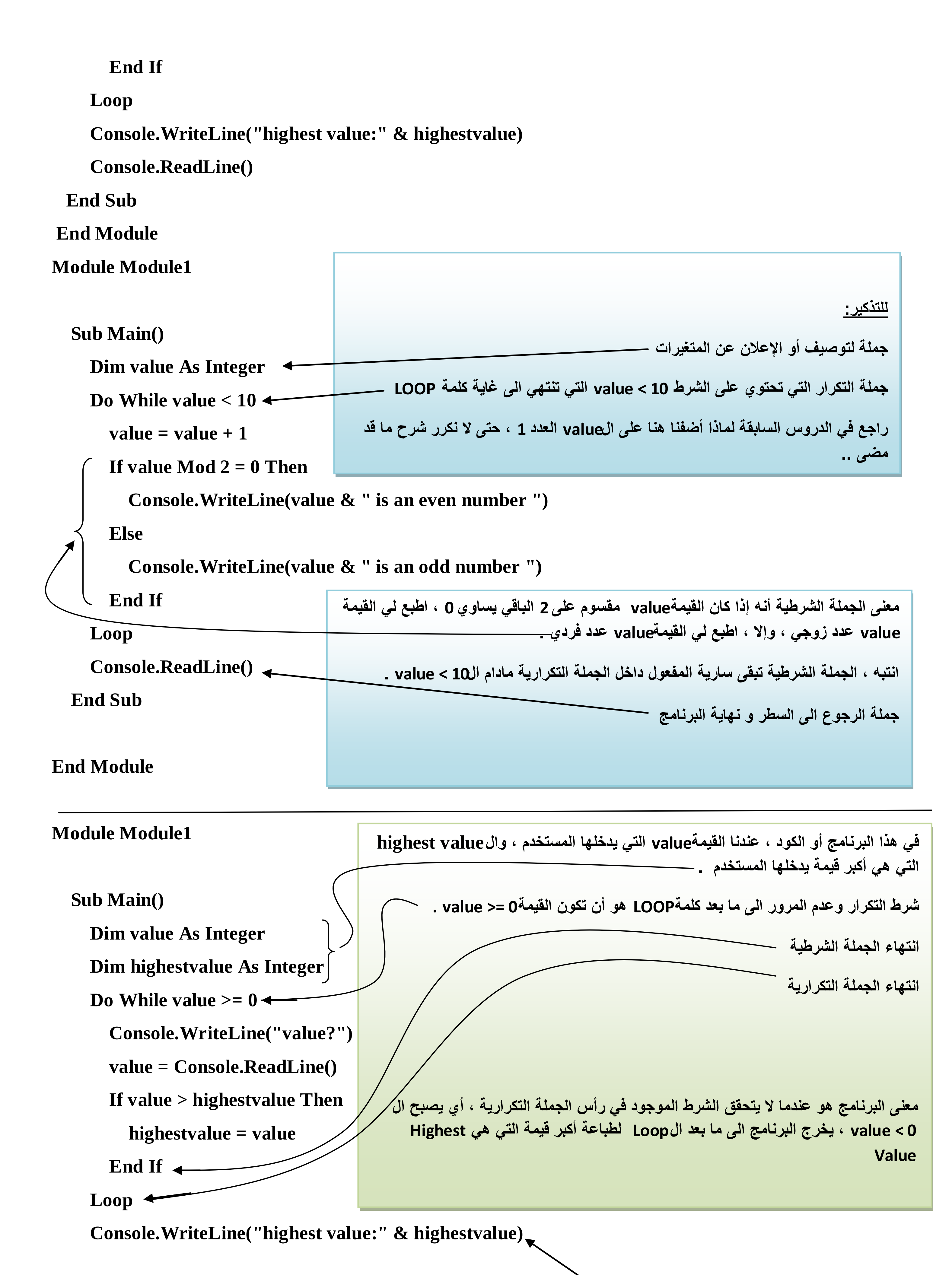
    Sub Main()
        Dim value As Integer
        Dim highestvalue As Integer
        Do While value >= 0
            Console.WriteLine("value?")
            value = Console.ReadLine()
            If value > highestvalue Then
                highestvalue = value
```

جمل لتوصيف المتغيرات

الجملة التكرارية : سوف يقوم بطباعة الأرقام التي يدخلها المستخدم شرط أن تكون أكبر أو يساوي 0 ، أرقام موجبة .

معنى هذا الشرط أن أكبر قيمة يدخلها المستخدم هي الhighest value . الذي سوف يطبعه البرنامج عند خروجه من الجملة التكرارية .

انتبه ، متى يطبع ال highest value ؟ الجواب : عندما يخرج من الجملة التكرارية أي مابعد ال loop لعدم تحقق شرط الجملة التكرارية وهو أن القيمة التي أدخلها المستخدم في الأخير سالبة أصغر من 0 .



```
        End If
      Loop
      Console.WriteLine("highest value:" & highestvalue)
      Console.ReadLine()
   End Sub
 End Module
```

```
Module Module1

    Sub Main()
      Dim value As Integer
      Do While value < 10
        value = value + 1
        If value Mod 2 = 0 Then
          Console.WriteLine(value & " is an even number ")
        Else
          Console.WriteLine(value & " is an odd number ")
        End If
      Loop
      Console.ReadLine()
    End Sub

End Module
```

<u>للتذكير:</u>

جملة لتوصيف أو الإعلان عن المتغيرات

جملة التكرار التي تحتوي على الشرط value < 10 التي تنتهي الى غاية كلمة LOOP

راجع في الدروس السابقة لماذا أضفنا هنا على الvalue العدد 1 ، حتى لا نكرر شرح ما قد مضى ..

معنى الجملة الشرطية أنه إذا كان القيمةvalue مقسوم على 2 الباقي يساوي 0 ، اطبع لي القيمة value عدد زوجي ، وإلا ، اطبع لي القيمةvalue عدد فردي .

انتبه ، الجملة الشرطية تبقى سارية المفعول داخل الجملة التكرارية مادام الvalue < 10 .

جملة الرجوع الى السطر و نهاية البرنامج

```
Module Module1

    Sub Main()
      Dim value As Integer
      Dim highestvalue As Integer
      Do While value >= 0
        Console.WriteLine("value?")
        value = Console.ReadLine()
        If value > highestvalue Then
          highestvalue = value
        End If
      Loop
      Console.WriteLine("highest value:" & highestvalue)
```

في هذا البرنامج أو الكود ، عندنا القيمةvalue التي يدخلها المستخدم ، والhighest value التي هي أكبر قيمة يدخلها المستخدم .

شرط التكرار وعدم المرور الى ما بعد كلمةLOOP هو أن تكون القيمة0 =< value .

انتهاء الجملة الشرطية

انتهاء الجملة التكرارية

معنى البرنامج هو عندما لا يتحقق الشرط الموجود في رأس الجملة التكرارية ، أي يصبح ال value < 0 ، يخرج البرنامج الى ما بعد الLoop لطباعة أكبر قيمة التي هي Highest Value

```
    Console.ReadLine()
  End Sub

End Module
```

هذه جملة الإخراج سينفذها البرنامج عندما يخرج من التكرار ، متى ؟ الجواب : عندما تكون القيمة المدخلة أصغر من 0 مثلا (1-)، وعلى ما تحتوي هذه الجملة؟ الجواب : طباعة (ال Highest Value : أكبر قيمة أدخلها المستخدم) .

في هذه الحلقة سوف نتعرف على مصطلح جديد في عالم شرط التكرار

سوف نقوم بشرح هذه الجمل التكرارية :

DO WHILE …….LOOP كرر بينما ..... اللوب

DO UNTIL ……..LOOP كرر حتى ....اللوب

الجملة الأخيرة تحتوي على شرط إنهاء التكرار وليس شرط التكرار .

لنأخذ تطبيقا لهذه الجملة التي سبق ذكرها :

```
Module Module1
  Sub Main()
    Dim value As Integer
    Do Until value >= 10
      value = value + 1
      Console.WriteLine(value)
    Loop
    Console.ReadLine()

  End Sub
End Module
```

برنامج يطبع الأرقام من 1 إلى 10

للتذكير:

- جملة توصيف المتغيرات التي سوف تدخل في المعادلات البرمجية ← Dim value As Integer
- جملة التكرار التي هي في حد ذاتها تحتوي حدا لإنهاء التكرار ← Do Until value >= 10
- جملة المعادلة ← value = value + 1
- جملة الطباعة أو الكتابة ← Console.WriteLine(value)
- جملة إنهاء البرنامج ← Console.ReadLine()

نصيحة مهمة : أخي الكريم حتى لا تنسى أو تختلط عليك الأمور لكثرة ورود مصطلحات جديدة راجع دروس الجمل التكرارية والشرطية بدقة .

في هذه الحلقة سوف نضيف مصطلحين في عالم الجمل التكرارية الشرطية

DO WHILE ……… LOOP

DO ………. LOOP WHILE ← ماهو الفرق ؟ الجواب : أن الشرط ينفذ هنا في نهاية التكرار

****

DO UNTIL ………. LOOP

DO ……….. LOOP UNTIL ← ماهو الفرق ؟ الجواب : أن الشرط ينفذ في نهاية التكرار

برنامج لطباعة الأرقام من 1 الى الرقم الذي أدخله المستخدم بشرط أن يكون أصغر أو يساوي الvalue .

```
Module Module1

    Sub Main()
        Console.WriteLine("highest value?")
        Dim highest As Integer = Console.ReadLine()
        Console.WriteLine("row:")
        Dim value As Integer
        Do
            value = value + 1
            Console.WriteLine(value)
        Loop Until value >= highest
        Console.ReadLine()

    End Sub

End Module
```

جملة طباعة تطبع النص التالي فيما معناه:(أدخل القيمة الأكبر؟) تطلب من المستخدم إدخال قيمة .

جملة توصف متغيرال highest مع إدخاله في نفس الوقت من طرف المستخدم

ملاحضة مهمة: في البداية لايوجد أي شرط ، هو موجود في آخر الجملة التكرارية ،

لو افترضنا أن المستخدم أدخل قيمة للhighest تساوي 0 ما ذا سيحصل ؟ سوف يطبع لنا البرنامج رقم 1 فقط ، كيف حصل ذالك ؟

1- البرنامج أضاف على القيمة المبدئية للvalue الذي هو 0 رقم 1.

2- فطبعها .

3- ثم نزل الى الجملة التكرارية ، التي تنهي التكرارعند تحقق الشرط ، فيقرأ أن رقم1 أكبر من القيمة 0 التي أدخلها المستخدم فينهي التكرار مادام الشرط قد تحقق (راجع شروح الجمل التكرارية – Until و Do while-)

<u>انتبــــــــــه</u> لو جعلنا الشرط في بداية الجملة ، ماذا سيحصل ؟ لن يطبع لنا أي رقم.

الكود موجود تحت هذه السطور.

```
Module Module1
   Sub Main()
        Console.WriteLine("highest value?")
        Dim highest As Integer = Console.ReadLine()
        Console.WriteLine("row:")
        Dim value As Integer
        Do Until value >= highest
            value = value + 1
            Console.WriteLine(value)
        Loop
        Console.ReadLine()
  End Sub
```

نفس الكود أعلاه لكن مع نقل الشرط الى بداية الجملة

لنقرأ الكود سطرا سطرا

توصيف المتغيرات مع ادخالها

القيمة المبدئية لل value وال highest هو 0

لنفترض أن المستخدم أدخل قيمة للhighest تساوي 0

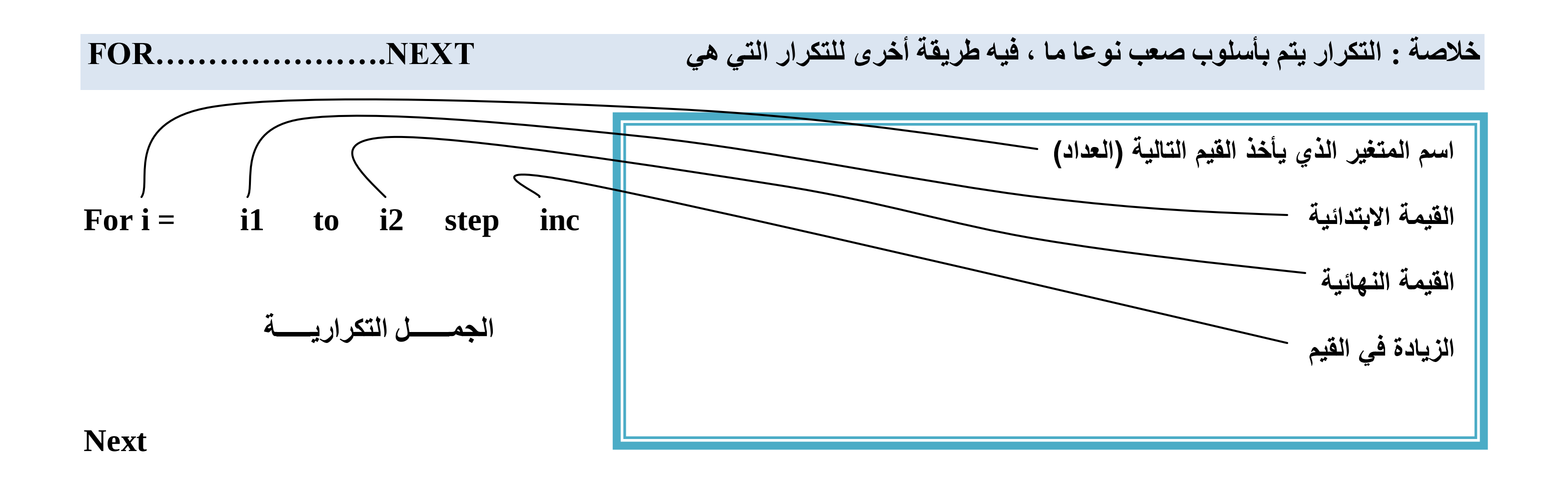
معنى الشرط : إذا كان الvalue الذي هو 0 أكبر أويساوي من ال highest الذي هو 0 فاجعل نهاية للتكرار وإيقاف للطباعة (لا أكرر شرح ما مضى فراجع الجمل التكرارية والشرطية كلما انتقلت إلى درس جديد أو دخل عليك مصطلح جديد) .

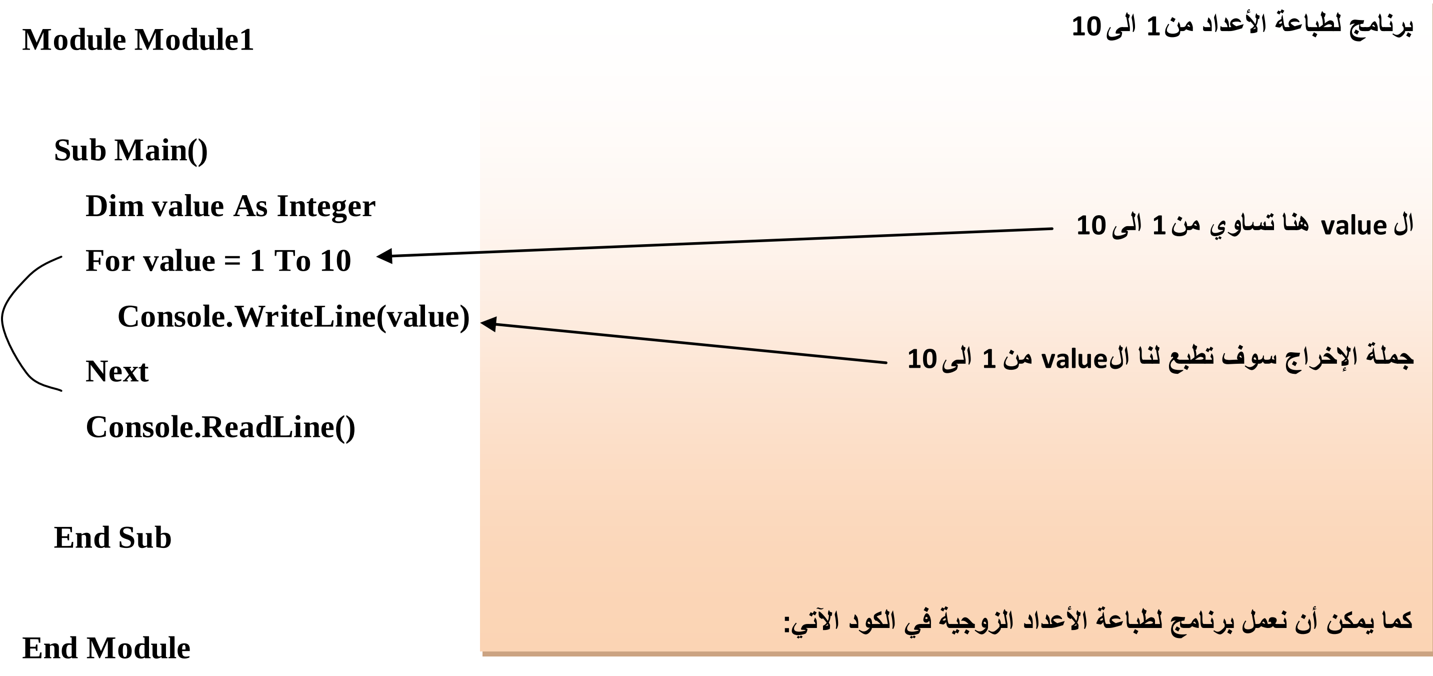
إذا هنا الشرط تحقق منذ البداية قبل أن ينزل إلى المعالجة أو المعادلة فلن يطبع لنا أي رقم .

سؤال من هو الكود الذي يفيدنا في تحقيق هدفنا ، الحالي أم السابق ؟

حسب رأيي الكود الحالي هو حل لمشكلة الكود الأول ، لأن ال0 الذي أدخله المستخدم لا يمكن أن يكون هو أكبر قيمة للأرقام التي يطبعها البرنامج .

End Module





```
Module Module1
  Sub Main()
     Dim value As Integer
     For value = 2 To 10 Step 2
        Console.WriteLine(value)
     Next
     Console.ReadLine()
 End Sub
End Module
```

هنا في هذا الكود استعملنا Step 2 أي أنه سوف يقوم بطباعة الأرقام بزيادة 2 ، الى غاية رقم 10 ، ابتداءا برقم 2 هكذا:

2

4

6

8

10

ويمكن أن نطبع الأرقام عكس ما سبق أي تنازليا كما هو مبين في الكود الآتي:

```
Module Module1

    Sub Main()
        Dim value As Integer
        For value = 10 To 0 Step -2
            Console.WriteLine(value)
        Next
        Console.ReadLine()

    End Sub

End Module
```

توصيف متغير صحيح value

العداد الذي سوف يكرر الطباعة من10 الى 0 بـ -2

هكذا تكون الطباعة:

10

8

6

4

2

0

```
Module Module1

    Sub Main()
        Console.WriteLine("Enter start value?")
        Dim startvalue As Integer = Console.ReadLine()
        Console.WriteLine("Enter end value?")
        Dim endvalue As Integer = Console.ReadLine()
        Console.WriteLine("Enter incremental value?")
        Dim incvalue As Integer = Console.ReadLine()
        Dim value As Integer
        For value = startvalue To endvalue Step incvalue
            Console.WriteLine(value)
        Next
        Console.ReadLine()

    End Sub

End Module
```

جمل لطباعة رسائل موجهة الى المستخدم

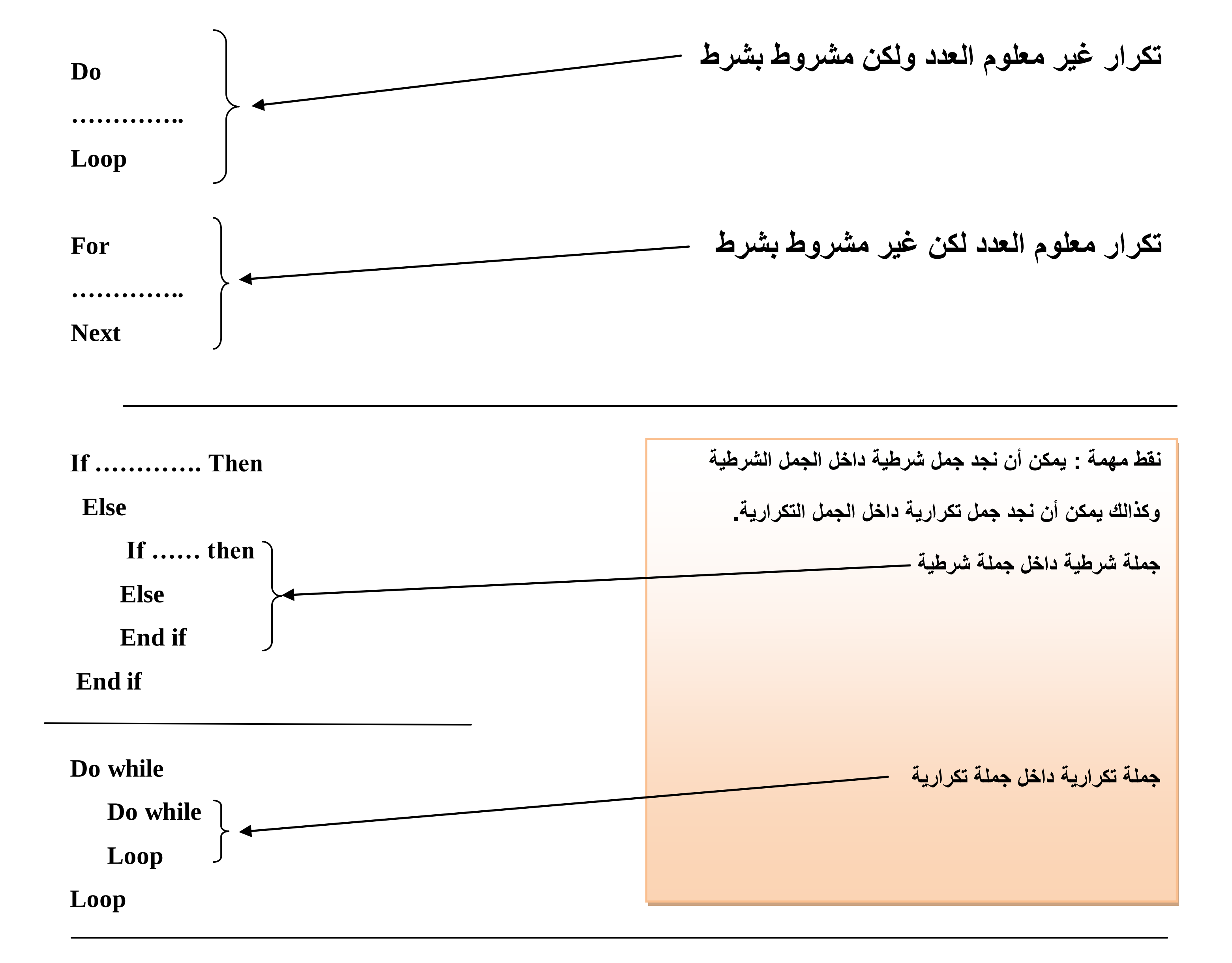
جمل توصيف المتغيرات والإدخال في آن واحد

توصيف متغير صحيح value العداد

معناه ، أن عداد الطباعة سيكون من الرقم الابتدائيStart value الى غاية الرقم النهائي End value اللذان أدخلهما المستخدم ، بزيادةIncvalue.

جملة لإنهاء البرنامج

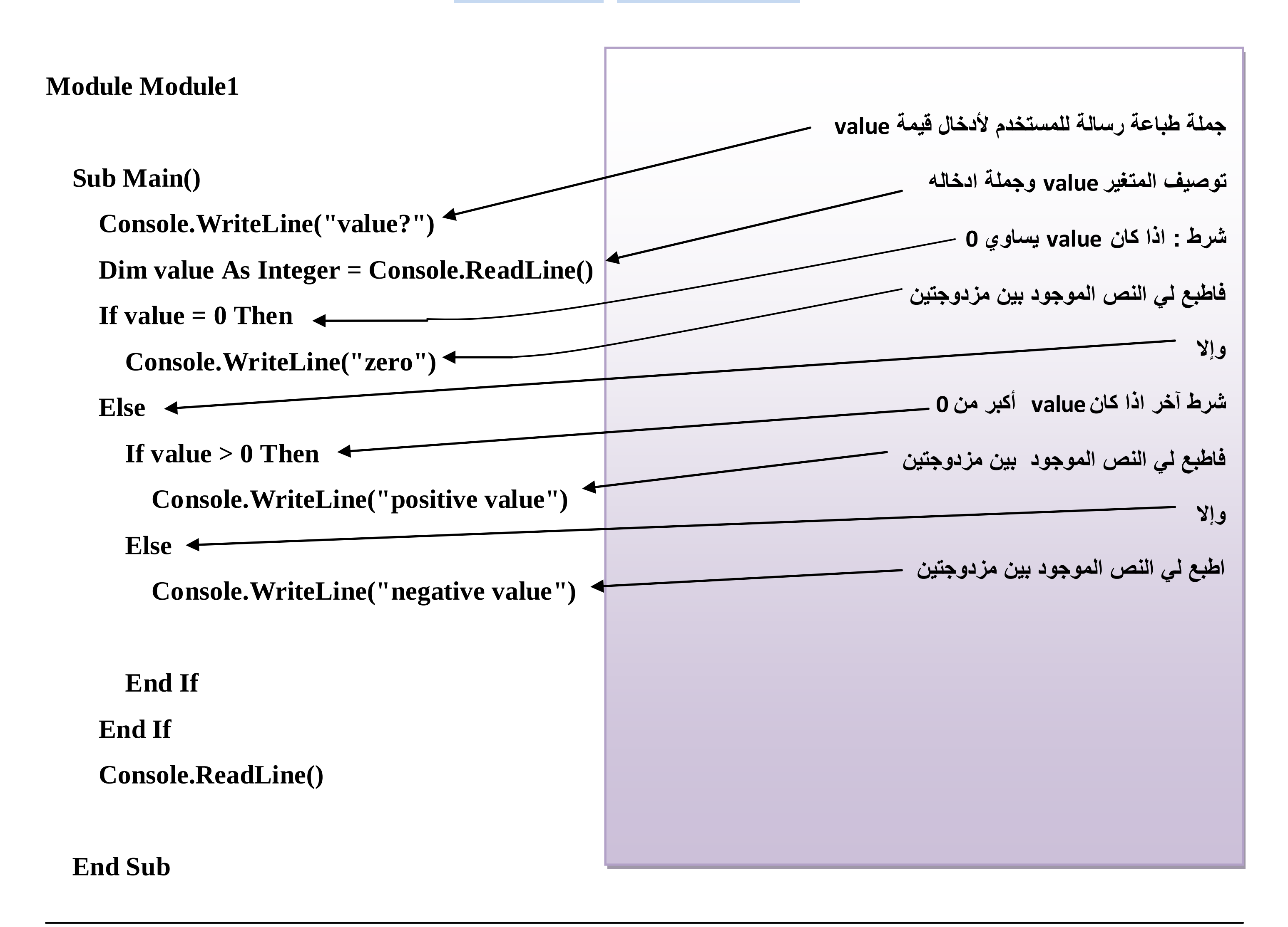
سؤال مهم : ما هو الفرق بين الجمل التكرارية ، **Do… Loop** و **For … Next** ؟



نهاية المحاضرة رقم 02

المحاضرة رقم 3

Selections الخيـــــــــــارات




```
Module Module1

    Sub Main()
        Console.WriteLine("value?")
        Dim value As Integer = Console.ReadLine()
        If value = 0 Then
            Console.WriteLine("zero")
        Else
            If value > 0 Then
                Console.WriteLine("positive value")
            Else
                Console.WriteLine("negative value")

            End If
        End If
        Console.ReadLine()

    End Sub
```

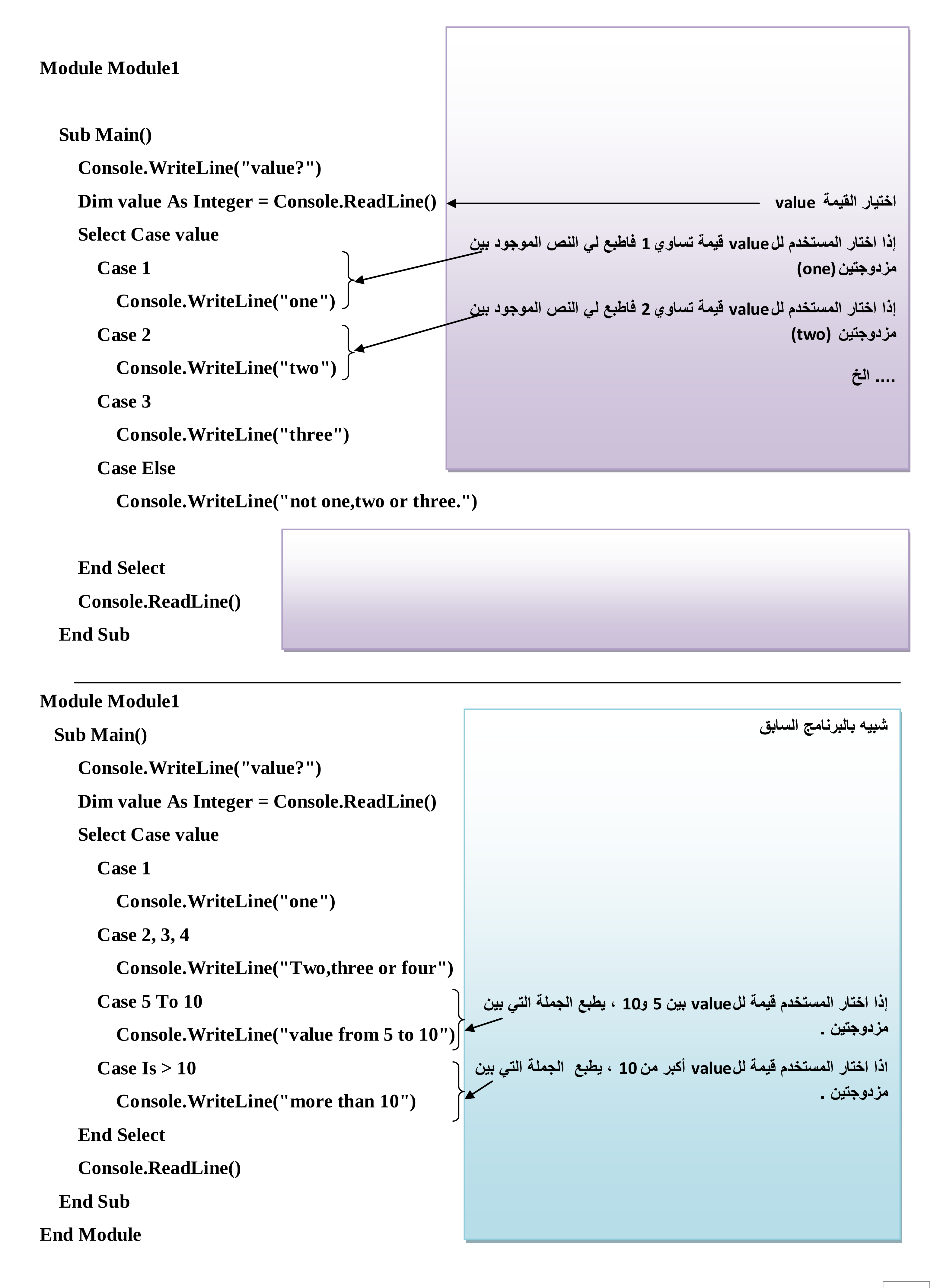
نكتب الكود السابق بصيغة أخرى

```
Module Module1

    Sub Main()
        Console.WriteLine("value?")
        Dim value As Integer = Console.ReadLine()
        If value = 0 Then Console.WriteLine("zero")
        If value > 0 Then Console.WriteLine("positive value")
        If value < 0 Then Console.WriteLine("negative value")
        Console.ReadLine()
    End Sub

End Module
```

لكن أفضل طريقة لكتابة البرامج المتعدد الشروط يأتي في البرنامج التالي:




```
Module Module1

  Sub Main()
    Console.WriteLine("value?")
    Dim value As Integer = Console.ReadLine()
    Select Case value
      Case 1
        Console.WriteLine("one")
      Case 2
        Console.WriteLine("two")
      Case 3
        Console.WriteLine("three")
      Case Else
        Console.WriteLine("not one,two or three.")

    End Select
    Console.ReadLine()
  End Sub
```

```
Module Module1
  Sub Main()
    Console.WriteLine("value?")
    Dim value As Integer = Console.ReadLine()
    Select Case value
      Case 1
        Console.WriteLine("one")
      Case 2, 3, 4
        Console.WriteLine("Two,three or four")
      Case 5 To 10
        Console.WriteLine("value from 5 to 10")
      Case Is > 10
        Console.WriteLine("more than 10")
    End Select
    Console.ReadLine()
  End Sub
End Module
```

```
Module Module1
  Sub Main()
      Console.WriteLine("value?")
      Dim value As Integer = Console.ReadLine()
      Select Case value
         Case Is > 10
            Console.WriteLine("more than 10")
         Case Is > 100
            Console.WriteLine("more than 100")
         Case Is > 1000
            Console.WriteLine("more than 1000")
      End Select
      Console.ReadLine()
   End Sub

End Module
```

هذا البرنامج كتابته صحيحة لكن عمله غير صحيح .

المشكلة هو أن البرنامج يطبع لنا دائما النص الموجود في الجملة الأولى (more than 10) وإن أدخل المستخدم قيمة أكبر من100 ، أو أكبر من 1000 ، لماذا ؟

لأن القيم التي أدخلها المستخدم هي أيضا أكبر من10 . يطبع النص الأول ثم ينزل الى End Select وينهي البرنامج .

الحل هو أن نعكس الترتيب من تحت لفوق

تابع

```
Module Module1
  Sub Main()
      Console.WriteLine("value?")
      Dim value As Integer = Console.ReadLine()
      Select Case value
        Case Is > 1000
            Console.WriteLine("more than 1000")
        Case Is > 100
            Console.WriteLine("more than 100")
        Case Is > 10
            Console.WriteLine("more than 10")
       End Select
      Console.ReadLine()
   End Sub
End Module
```

الآن البرنامج يستطيع أن يطبع أي نص حسب القيمة المختارة التي يدخلها المستخدم .بدون مشكلة.

ملاحضة جد جد مهمة :

الترتيب في هذه الحالة مهم جدا ، لماذا ؟

لأن القيمة التي هي أكبر من10 تشمل أيضا القيمة التي هي أكبر من100 و القيمة التي هي أكبر من1000، وكذالك القيمة التي هي أكبر من100 تشمل أيضا القيمة التي هي أكبر من1000، بمعنى دائما نرتب الشروط من العام الى الأعم .

```
Module Module1

  Sub Main()
    Console.WriteLine("value?")
    Dim value As Integer = Console.ReadLine()
    Dim info As String
    Select Case value
      Case Is > 100
        info = "more than 100"
      Case Is > 50
        info = "more than 50"
      Case 1, 11
        info = "one or eleven"
      Case 2 To 50
        info = "number from 2 to 50"
      Case 0
        info = "zero"
      Case -50 To -1
        info = "Number from -50 to -1"
      Case Is < -100
        info = "less than -100"
    End Select
    Console.WriteLine(info)
    Console.ReadLine()
  End Sub

End Module
```

ملاحضة : الفرق بين هذا الكود والكود السابق هو تموقع جملة الطباعة ، في الكود السابق كانت تأتي مباشرة بعد شرط من الشروط ، في هذا الكود وضعت متأخرا بعد الكلمة الأخيرة للجملة الشرطية التسلسلية End select

في هذا الكود أدخلنا متغير جديدinfo من نوع نصي ليحمل النص حسب كل شرط من الشروط المتسلسلة ، ثم بعد ذالك تستدعيه جملة الطباعة المتواجدة في آخر الكود .

جملة الإخراج أو الطباعة ← Console.WriteLine(info)

ARRAYS

المصفوفات

نقوم بعمل على أساس المخرجات التالية :

المخرجات هي الأرقام المكتوبة بالأحمر

سوف يطبع لنا البرنامج العملية الحسابية التالية:

60=30+20+10

value 1 ? 10

Value 2 ? 20

value 3 ? 30

```
Module Module1

    Sub Main()
        Dim value1, value2, value3 As Integer
        Dim sum As Integer
        Console.WriteLine("value1?")
        value1 = Console.ReadLine()
        sum += value1
        Console.WriteLine("value2?")
        value2 = Console.ReadLine()
        sum += value2
        Console.WriteLine("value3?")
        value3 = Console.ReadLine()
        sum += value3
        Console.Write(value1)
        Console.Write("+")
        Console.Write(value2)
        Console.Write("+")
        Console.Write(value3)
        Console.Write("=")
        Console.Write(sum)
        Console.ReadLine()

    End Sub

End Module
```

هذه الجملة معناها بصيغة أخرى sum = sum+ value1

دائما القيمة الأولية لأي متغير هي0

لاحظ أننا كتبنا write وليس writeline الأولى نطبع في نفس السطر والثانية الرجوع الى سطر جديد

هذا البرنامج أو المشروع كتب بطريقة صعبة ، علينا التفكير في طريقة أسهل من هذا

الحل في البرنامج ما بعد الآتي :

```
Module Module1

    Sub Main()
        Dim values(2) As Integer
        Dim sum As Integer
        Console.WriteLine("value1?")
        values(0) = Console.ReadLine()
        sum += values(0)
        Console.WriteLine("value2?")
        values(1) = Console.ReadLine()
        sum += values(1)
        Console.WriteLine("value3?")
        values(2) = Console.ReadLine()
        sum += values(2)
        Console.Write(values(0))
        Console.Write("+")
        Console.Write(values(1))
        Console.Write("+")
        Console.Write(values(2))
        Console.Write("=")
        Console.Write(sum)
        Console.ReadLine()

    End Sub

End Module
```

سوف نطور هذا البرنامج ونتفادى التكرار الذي تلاحظونه واضحا بالجملة التكرارية For Next

---

```
Module Module1
    Sub Main()

        Dim values(2) As Integer
        Dim sum As Integer
        Dim index As Integer

        For index = 0 To 2
            Console.WriteLine("value" & index + 1 & "?")
            values(index) = Console.ReadLine()
            sum += values(index)
        Next

        Console.Write(values(0))

        For index = 1 To 2
            Console.Write("+")
            Console.Write(values(index))
        Next

        Console.Write("=")
        Console.Write(sum)
        Console.ReadLine()

    End Sub
End Module
```

```
Module Module1
    Sub Main()

        Console.WriteLine("count of values?")

        Dim count As Integer = Console.ReadLine()
        Dim values(count) As Integer
        Dim sum As Integer
        Dim index As Integer

        For index = 0 To count - 1
            Console.WriteLine("value" & index + 1 & "?")
            values(index) = Console.ReadLine()
            sum += values(index)
        Next

        Console.Write(values(0))

        For index = 1 To count - 1
            Console.Write("+")
            Console.Write(values(index))
        Next

        Console.Write("=")
        Console.Write(sum)
        Console.ReadLine()

    End Sub
End Module
```

لماذا كتبنا هنا ناقص واحد ؟ لأن القيمة الأولية لأي متغير أو مصفوفة هو **0** ، والصفر يعتد به واحد ، إذا أدخل المستخدم مثلا رقم**10** يصبح **11**

كتابة الأرقام بطريقة تنازلية

```
Module Module1
    Sub Main()

        Dim count As Integer = 5
        Dim upperbound As Integer = count - 1
        Dim values(upperbound) As Integer
        Dim index As Integer

        For index = 0 To upperbound
            Console.Write(" value " & (index + 1) & " : ")
            values(index) = Console.ReadLine()
        Next

        For index = upperbound To 0 Step -1
            Console.WriteLine(" Element " & (index + 1) & "(at index " &
(index) & " ): " & values(index))

        Next

        Console.ReadLine()

    End Sub
End Module
```

Step-1 طباعة الأرقام بخطوة ناقص 1

```
Module Module1
    Sub Main()

        Console.Write(" count: ")

        Dim count As Integer = Console.ReadLine()
        Dim upperbound As Integer = count - 1
        Dim values(upperbound) As Integer
        Dim index, highestvalueindex As Integer

        For index = 0 To upperbound
            Console.Write(" value " & (index + 1) & " : ")
            values(index) = Console.ReadLine()
            If values(highestvalueindex) < values(index) Then
                highestvalueindex = index
            End If
        Next

        Console.WriteLine(" highest value: " & values(highestvalueindex) &
"(Element " & (highestvalueindex + 1) & " at index " & highestvalueindex & ")")

        Console.ReadLine()

    End Sub
End Module
```

نـنجز بـرنـامـجا من مصفوفة ARRAY تـتكون من 10 عناصر Elements الـتي تـخزن قـيما values من 10 الـى 1 .

Index : 0 1 2 3 4 5 6 7 8 9

Index : 10 9 8 7 6 5 4 3 2 1

بـحيث أن الـبرنـامـج سوف يطبع لـنا كـالـتالـي :

Element1 (index0) : 10

Element2 (index1) : 9

Element2 (index1) : 8

Element2 (index1) : 7

Element2 (index1) : 6

Element2 (index1) : 5

Element2 (index1) : 4

……………………etc

```
Module Module1
    Sub Main()

        Dim values(9) As Integer
        Dim index As Integer

        For index = 0 To 9
            values(index) = 10 - index
            Console.WriteLine("Element" & (index + 1) & " ( index " & index & "
):" & values(index))
        Next

        Console.ReadLine()

    End Sub
End Module
```

نـنجز بـرنـامـجا يـرتـب مصفوفة ARRAY تـحتـوي عـلى 10 عناصر ELEMENTS تـخزن قـيما من 1 الـى 10

Index 0 1 2 3 4 5 6 7 8 9

Value 1 2 3 4 5 6 7 8 9 10

وفـي تـريـب ثـانـي بـشكل عكسي

Index 0 1 2 3 4 5 6 7 8 9

Value 10 9 8 7 6 5 4 3 2 1

```
Module Module1
    Sub Main()

        Dim count As Integer = 10
        Dim upperbound As Integer = count - 1
        Dim values(upperbound) As Integer
```

```
        For index = 0 To upperbound
            values(index) = index + 1
        Next

        For index = 0 To upperbound
            Console.WriteLine(" Element " & (index + 1) & " (index " & index &
") : " & values(index))
        Next

        Console.WriteLine()
        Dim backup As Integer

        For index = 0 To (count / 2) - 1
            backup = values(index)
            values(index) = values(upperbound - index)
            values(upperbound - index) = backup
        Next

        For index = 0 To upperbound
            Console.WriteLine(" Element " & (index + 1) & " (index " & index &
") : " & values(index))
        Next

        Console.ReadLine()

    End Sub
End Module
```

هذه الجملة معناها الرجوع الى سطر وتركه فارغا

ننجز برنامجا يرتب مصفوفة ARRAY تحتوي على 10 عناصر ELEMENTS تخزن قيما من 1 الى 10

المسألة التي أكتبها هنا مختصرة عندي ، فانظر في الفيديو هي بشكل أوضح وأعم

Index 0 1 2 3 4 5 6 7 8 9

Value 1 2 3 4 5 6 7 8 9 10

سوف نزحزح القيم برقم واحد الى اليسار

Index 0 1 2 3 4 5 6 7 8 9

Value 2 3 4 5 6 7 8 9 10 1

```
Module Module1
    Sub Main()

        Dim count As Integer = 10
        Dim upperbound As Integer = count - 1
        Dim values(upperbound) As Integer

        For index = 0 To upperbound
            values(index) = index + 1
        Next

        For index = 0 To upperbound
```

```
            Console.WriteLine(" Element " & (index + 1) & " (index " & index & ") : " & values(index))
        Next

        Console.WriteLine()
        Dim backup As Integer = values(0)

        For index = 0 To (upperbound - 1)
            values(index) = values(index + 1)
        Next

        values(upperbound) = backup

        For index = 0 To upperbound
            Console.WriteLine(" Element " & (index + 1) & " (index " & index & ") : " & values(index))
        Next

        Console.ReadLine()

    End Sub
End Module
```

نـنجز بـرنـامـجا يـرتـب ثـلاث مصفـوفـاتARRAYs تـحتـوي عـلى 10 عـناصر ELEMENTS تـخزن قـيمـا من 1 الـى 10

هذه المقدمة أو المسألة مختصرة ، ارجع الى الفيديو

الـمصفـوفـة 1 قـيم من 1 الـى 10:

Index 0 1 2 3 4 5 6 7 8 9

Value 1 2 3 4 5 6 7 8 9 10

الـمصفـوفـة2 قـيم من 10 الـى 1

Index 0 1 2 3 4 5 6 7 8 9

Value10 9 8 7 6 5 4 3 2 1

الـمصفـوفـة3 قـيم بـالـشكل الـتـالـي

Index 0 1 2 3 4 5 6 7 8 9

Value10 9 8 7 6 6 7 8 9 10

```
Module Module1

    Sub Main()
        Dim count As Integer = 10
        Dim upperbound As Integer = count - 1
        Dim values1(upperbound) As Integer
        Dim values2(upperbound) As Integer
        Dim values3(upperbound) As Integer
        Dim index As Integer

        For index = 0 To upperbound
            values1(index) = index + 1
        Next

        Console.WriteLine("Array1: ")

        For index = 0 To upperbound
```

```
            Console.WriteLine("Element " & (index + 1) & "(index " & index &
"): " & values1(index))
        Next

        Console.WriteLine()

        For index = 0 To upperbound
            values2(index) = count - index
        Next

        Console.WriteLine("Array2: ")

        For index = 0 To upperbound
            Console.WriteLine("Element " & (index + 1) & "(index " & index &
"): " & values2(index))
        Next

        Console.WriteLine()

        For index = 0 To upperbound
            If values1(index) >= values2(index) Then
                values3(index) = values1(index)
            Else
                values3(index) = values2(index)
            End If
        Next

        Console.WriteLine("Array3: ")

        For index = 0 To upperbound
            Console.WriteLine("Element " & (index + 1) & "(index " & index &
"): " & values3(index))
        Next

        Console.WriteLine()
        Console.ReadLine()
    End Sub

End Module
```

ننجز برنامجا بمصفوفة بحيث عناصرها أرقام على شكل **Fibonacci** أي أن الرقم الذي يلي هو مجموع الرقمين السابقين له

Fibonacci : 1 1 2 3 5 8 13 21 34 55 89 ……..

```
Module Module1

    Sub Main()
        Console.Write("count: ")
        Dim count As Integer = Console.ReadLine()

        If count <= 0 Then
            Console.WriteLine("Error minimum count1. ")
        Else
            Dim upperbound As Integer = count - 1
            Dim fibonacci(upperbound) As Integer
            Dim index As Integer
            fibonacci(0) = 1
```

```
            If count > 1 Then
                fibonacci(1) = 1

                For index = 2 To upperbound
                    fibonacci(index) = fibonacci(index - 1) + fibonacci(index -
2)
                Next
            End If

            For index = 0 To upperbound
                Console.Write(fibonacci(index) & " ")
            Next
        End If

        Console.ReadLine()
    End Sub

End Module
```

رقم 4 هو جداء الرقمين 2 في العمود الثاني و2 في السطر الثاني ، 12 هو جداء الرقمين 4 في العمود الرابع والرقم 3 في السطر الثالث وهكذا.

سوف نقوم بعمل برنامج ينتج لنا جدول على هذا الشكل:

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24

```
Module Module1

    Sub Main()
        Console.Write("Row count: ")
        Dim rowcount As Integer = Console.ReadLine()
        Console.Write("column count: ")
        Dim columncount As Integer = Console.ReadLine()
        Dim values(rowcount, columncount) As Integer
        Dim i, j As Integer

        For i = 1 To rowcount
            For j = 1 To columncount
                values(i, j) = i * j
            Next
        Next
        Console.WriteLine("*************")
        For i = 1 To rowcount
            For j = 1 To columncount
                Console.Write(values(i, j))
                Console.Write("  ")
            Next
            Console.WriteLine("  ")
        Next
        Console.ReadLine()
    End Sub

End Module
```

نهاية المحاضرة رقم 3

بداية المحاضرة رقم 4

FUNCTIONS AND SUBROUTINES

الدوال و البرامج الفرعية

في الفيجوال بيسك دوت نت دوال كثيرة كالدوال الرياضية : sin , cos , tang... ودوال التعامل مع النصوص : len , trim , ucase ... ودوال أخرى تجدها مصنفة أو بالألفبائية في الكتب الخاصة.

في البرنامج الآتي سوف نقوم بعملية رياضية أي حسابية : برنامج للفاكتوريال

$$c_m^n = \frac{n!}{m!(n-m)!}$$

```
Module Module1

    Sub Main()
        Console.Write("Enter the first number: ")
        Dim n As Integer = Console.ReadLine()
        Console.Write("Enter the second number: ")
        Dim m As Integer = Console.ReadLine()
        Dim nm, Facn, Facm, Facnm, combin, i As Integer

        Facn = 1
        For i = 2 To n
            Facn = Facn * i
        Next

        Facm = 1
        For i = 2 To m
            Facm = Facm * i
        Next

        nm = n - m

        Facnm = 1
        For i = 2 To nm
            Facnm = Facnm * i
        Next
        combin = Facn / (Facm * Facnm)
        Console.WriteLine("c(" & n & "," & m & ") = " & combin)
        Console.ReadLine()

    End Sub

End Module
```

سوف نستخدم هذا البرنامج بطريقة أفضل بواسطة الدوال Functions لأننا كما لاحظتم كررنا الكلمة Fac الفكتوريال عدة مرات . الطريقة كما سيأتي في البرنامج الآتي هو استعمال دالة نسميها Factorial لتجنب التكرار

```
Module Module1

    Sub Main()
        Console.Write("Enter the first number: ")
        Dim n As Integer = Console.ReadLine()
        Console.Write("Enter the second number: ")
        Dim m As Integer = Console.ReadLine()
```

```
        Dim combin As Integer = Factorial(n) / Factorial(m) * Factorial(n - m)
        Console.WriteLine("c(" & n & "," & m & ") = " & combin)
        Console.ReadLine()
    End Sub
    Function Factorial(ByVal j As Integer)

        Dim k, sum As Integer
        sum = 1
        For k = 2 To j
            sum = sum * k
        Next
        Return (sum)

    End Function

End Module
```

برنامج شبيه تقريبا للبرنامج السابق نتدرب فيه على صنع واستخدام الدالات

```
Module Module1

    Sub Main()
        Console.Write("Enter the number: ")
        Dim n As Integer = Console.ReadLine()
        Dim Fn As Integer
        Fn = sumpower(n, 4) / sumpower(n, 2) * sumpower(n, 3)
        Console.WriteLine("F(" & n & ")= " & Fn)
        Console.ReadLine()
    End Sub

 Function sumpower(ByVal j As Integer, ByVal p As Integer)

        Dim k, sum As Integer
        For k = 1 To j
            sum += k ^ p
        Next
        Return (sum)

    End Function

End Module
```

$$F(n) = \frac{\sum_{i=1}^{n} i^4}{\sum_{i=1}^{n} i^2 * \sum_{i=1}^{n} i^3}$$

في البرنامجين السابقين استخدمنا الدالة FUNCTION والآن تعالوا نستخدم البرنامج الفرعي SUBROUTINE (SUB) لنلاحظ الفرق بينهما

ونأخذ نفس المشروع السابق الفاكتوريال FAC :

$$c_m^n = \frac{n!}{m!(n-m)!}$$

```
Module Module1

    Sub Main()
        Console.Write("Enter the first number: ")
```

```
        Dim n As Integer = Console.ReadLine()
        Console.Write("Enter the second number: ")
        Dim m As Integer = Console.ReadLine()
        Dim nm, Facn, Facm, Facnm, combin, i As Integer
        Factorial(n, Facn)
        Factorial(m, Facm)
        Factorial(n - m, Facnm)
        combin = Facn / (Facm * Facnm)
        Console.WriteLine("c(" & n & "," & m & ") = " & combin)
        Console.ReadLine()
    End Sub
    Sub Factorial(ByVal j As Integer, ByRef sum As Integer)

        Dim k As Integer
        sum = 1

        For k = 2 To j
            sum = sum * k
        Next

    End Sub

End Module
```

أخي الكريم ، في كل بداية محاضرة ونهايتها، يقوم الدكتور عاطف العراقي بمراجعة شاملة تحتوي على المعلومات الهامة والأساسية التي ينبغي على المتعلم التعرف عليها .

فلا تمر عليها مر الكرام بل اعطها اهتمامك وبالك حتى ترسخ في ذهنك نظرا لأهميتها.

تابع نهاية المحاضرة ستجد الأجوبة عن الأسئلة التالية :

ما فائدة الدالة Fac والبرامج الفرعية sub ؟

ما الفارق بينهما ؟

لعبة مع المستخدم: نعم لا

Public Class Form1

Private Sub Button2_MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Button2.MouseMove

Button2.Text = "لا"

Button1.Text = "نعم"

End Sub

Private Sub Button1_MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Button1.MouseMove

Button1.Text = "لا"

Button2.Text = "نعم"

End Sub

End Class

==

عمليات حسابية بأزرار

Public Class Form1

```
  Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        addbtn.Text = "جمع"
        subbtn.Text = "طرح"
        multbtn.Text = "ضرب"
        dividbtn.Text = "قسمة"
        resultlbl.Text = ""
    End Sub

    Private Sub addbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
addbtn.Click
        resultlbl.Text = firstnumbertxt.Text + " + " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) + Val(secondnumbertxt.Text))
    End Sub

    Private Sub subbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
subbtn.Click
        resultlbl.Text = firstnumbertxt.Text + " - " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) - Val(secondnumbertxt.Text))
    End Sub

    Private Sub multbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles multbtn.Click
        resultlbl.Text = firstnumbertxt.Text + " * " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) * Val(secondnumbertxt.Text))
    End Sub

    Private Sub dividbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles dividbtn.Click
        resultlbl.Text = firstnumbertxt.Text + " / " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) / Val(secondnumbertxt.Text))
    End Sub
End Class
```

==

combobox عمليات حسابية بدون أزرار

```
Public Class Form1
```

```
 Private Sub operationcombo_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles operationCombo.SelectedIndexChanged

        Select Case operationCombo.Text

            Case "جمع"

                resultlbl.Text = firstnumbertxt.Text + " + " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) + Val(secondnumbertxt.Text))

            Case "طرح"

                resultlbl.Text = firstnumbertxt.Text + " - " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) - Val(secondnumbertxt.Text))

            Case "ضرب"

                resultlbl.Text = firstnumbertxt.Text + " * " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) * Val(secondnumbertxt.Text))

            Case "قسمة"

                resultlbl.Text = firstnumbertxt.Text + " / " + secondnumbertxt.Text + " = " +
Str(Val(firstnumbertxt.Text) / Val(secondnumbertxt.Text))

     End Select

    End Sub

End Class
```

===

حساب عدد حروف جملة وقلب حروفها

```
Public Class Form1


    Private Sub statmenttxt_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles statmenttxt.TextChanged

        charactnumlbl.Text = Len(statmenttxt.Text)

        riverstatmtlbl.Text = ""

        Dim i As Integer

        For i = Len(statmenttxt.Text) To 1 Step -1

            riverstatmtlbl.Text = riverstatmtlbl.Text + Mid(statmenttxt.Text, i, 1)


        Next

    End Sub

End Class
```

===

برنامج الآلة الحاسبة

```
Public Class Form1

   Dim firstnum As Decimal

   Dim operationname As String

   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

      screentxt.Text = "0"

   End Sub

 Private Sub btn0_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn0.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "0"

      Else

         screentxt.Text = screentxt.Text + "0"

      End If

   End Sub

 Private Sub btn1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn1.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "1"

      Else

         screentxt.Text = screentxt.Text + "1"

      End If

   End Sub

 Private Sub btn2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn2.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "2"

      Else

         screentxt.Text = screentxt.Text + "2"

      End If

   End Sub

Private Sub btn3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn3.Click
```

```
        If screentxt.Text = "0" Then
            screentxt.Text = "3"
        Else
            screentxt.Text = screentxt.Text + "3"
        End If
    End Sub
 Private Sub btn4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn4.Click
        If screentxt.Text = "0" Then
            screentxt.Text = "4"
        Else
            screentxt.Text = screentxt.Text + "4"
        End If
    End Sub
 Private Sub btn5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn5.Click
        If screentxt.Text = "0" Then
            screentxt.Text = "5"
        Else
            screentxt.Text = screentxt.Text + "5"
        End If
    End Sub
 Private Sub btn6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn6.Click
        If screentxt.Text = "0" Then
            screentxt.Text = "6"
        Else
            screentxt.Text = screentxt.Text + "6"
        End If
    End Sub
Private Sub btn7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn7.Click
        If screentxt.Text = "0" Then
```

```
        screentxt.Text = "7"

      Else

        screentxt.Text = screentxt.Text + "7"

      End If

    End Sub

Private Sub btn8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn8.Click

      If screentxt.Text = "0" Then

        screentxt.Text = "8"

      Else

        screentxt.Text = screentxt.Text + "8"

      End If

    End Sub

 Private Sub btn9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn9.Click

      If screentxt.Text = "0" Then

        screentxt.Text = "9"

      Else

        screentxt.Text = screentxt.Text + "9"

      End If

    End Sub

 Private Sub pointbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pointbtn.Click

      screentxt.Text = screentxt.Text + "."

    End Sub

 Private Sub addbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles addbtn.Click

      firstnum = Val(screentxt.Text)

      screentxt.Text = "0"

      operationname = "add"

    End Sub

Private Sub subbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles subbtn.Click

      firstnum = Val(screentxt.Text)
```

```
        screentxt.Text = "0"

        operationname = "sub"

    End Sub

Private Sub multbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
multbtn.Click

        firstnum = Val(screentxt.Text)

        screentxt.Text = "0"

        operationname = "mult"

    End Sub

 Private Sub dividbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
dividbtn.Click

        firstnum = Val(screentxt.Text)

        screentxt.Text = "0"

        operationname = "divid"

    End Sub

 Private Sub minnerbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles minnerbtn.Click

        screentxt.Text = Str(Val(screentxt.Text) * -1)

    End Sub

  Private Sub equalbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles equalbtn.Click

        Select Case operationname

            Case "add"

                screentxt.Text = Str(firstnum + Val(screentxt.Text))

            Case "sub"

                screentxt.Text = Str(firstnum - Val(screentxt.Text))

            Case "mult"

                screentxt.Text = Str(firstnum * Val(screentxt.Text))

            Case "divid"

                screentxt.Text = Str(firstnum / Val(screentxt.Text))

        End Select

 End Sub
```

```
   Private Sub clearbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles clearbtn.Click

      screentxt.Text = "0"

End Sub

 Private Sub lastnumclearbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lastnumclearbtn.Click

      screentxt.Text = Mid(screentxt.Text, 1, Len(screentxt.Text) - 1)

   End Sub

End Class
```

==

لقد أضفنا برنامج فرعي على الكود السابق الا أنه لم يشتغل بشكل جيد subroutine : operationexec()

```
Public Class Form1

   Dim firstnum As Decimal

   Dim operationname As String

   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

      screentxt.Text = "0"

   End Sub

 Private Sub btn0_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn0.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "0"

      Else

         screentxt.Text = screentxt.Text + "0"

      End If

   End Sub

 Private Sub btn1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn1.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "1"

      Else
```

```
        screentxt.Text = screentxt.Text + "1"

      End If

   End Sub

Private Sub btn2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn2.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "2"

      Else

         screentxt.Text = screentxt.Text + "2"

      End If

   End Sub

Private Sub btn3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn3.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "3"

      Else

         screentxt.Text = screentxt.Text + "3"

      End If

   End Sub

Private Sub btn4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn4.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "4"

      Else

         screentxt.Text = screentxt.Text + "4"

      End If

   End Sub

Private Sub btn5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn5.Click

      If screentxt.Text = "0" Then

         screentxt.Text = "5"

      Else

         screentxt.Text = screentxt.Text + "5"
```

```
        End If

    End Sub

 Private Sub btn6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn6.Click

        If screentxt.Text = "0" Then

            screentxt.Text = "6"

        Else

            screentxt.Text = screentxt.Text + "6"

        End If

    End Sub

Private Sub btn7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn7.Click

        If screentxt.Text = "0" Then

            screentxt.Text = "7"

        Else

            screentxt.Text = screentxt.Text + "7"

        End If

    End Sub

Private Sub btn8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn8.Click

        If screentxt.Text = "0" Then

            screentxt.Text = "8"

        Else

            screentxt.Text = screentxt.Text + "8"

        End If

    End Sub

Private Sub btn9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn9.Click

        If screentxt.Text = "0" Then

            screentxt.Text = "9"

        Else

            screentxt.Text = screentxt.Text + "9"

        End If
```

```
 End Sub

Private Sub pointbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles pointbtn.Click

        screentxt.Text = screentxt.Text + "."

    End Sub

Private Sub addbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles addbtn.Click

        operationexec()

        firstnum = Val(screentxt.Text)

        screentxt.Text = "0"

        operationname = "add"

    End Sub

Private Sub subbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles subbtn.Click

        operationexec()

        firstnum = Val(screentxt.Text)

        screentxt.Text = "0"

        operationname = "sub"

    End Sub

Private Sub multbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles multbtn.Click

        operationexec()

        firstnum = Val(screentxt.Text)

        screentxt.Text = "0"

        operationname = "mult"

    End Sub

Private Sub dividbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles dividbtn.Click

        operationexec()

        firstnum = Val(screentxt.Text)

        screentxt.Text = "0"

        operationname = "divid"

    End Sub
```

```
Private Sub minnerbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles minnerbtn.Click

        screentxt.Text = Str(Val(screentxt.Text) * -1)

    End Sub

Private Sub equalbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
equalbtn.Click

        operationexec()

    End Sub

Private Sub clearbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
clearbtn.Click

        screentxt.Text = "0"

 End Sub

 Private Sub lastnumclearbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lastnumclearbtn.Click

        screentxt.Text = Mid(screentxt.Text, 1, Len(screentxt.Text) - 1)

    End Sub


Sub operationexec()

        Select Case operationname

            Case "add"

                screentxt.Text = Str(firstnum + Val(screentxt.Text))

            Case "sub"

                screentxt.Text = Str(firstnum - Val(screentxt.Text))

            Case "mult"

                screentxt.Text = Str(firstnum * Val(screentxt.Text))

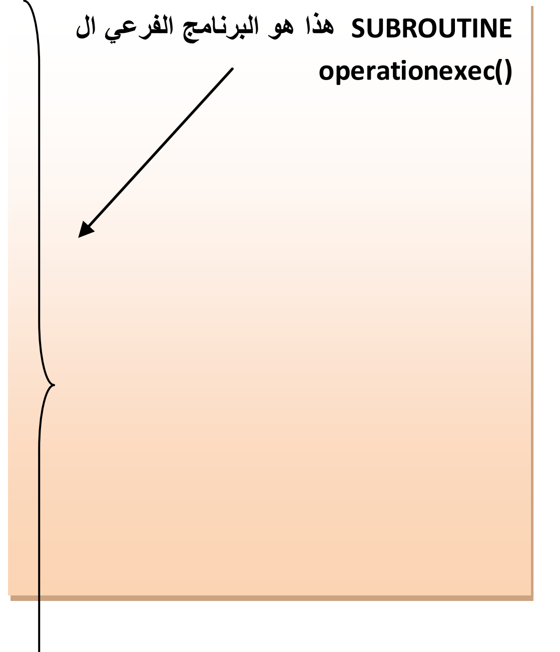
            Case "divid"

                screentxt.Text = Str(firstnum / Val(screentxt.Text))

        End Select

 End Sub

End Class
```



برنامـــج ساعـــة ميقاتيـــة

```
Public Class Form1
   Dim t1 As Decimal
   Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
       t1 += 0,1
       screenlbl.Text = Str(t1)
   End Sub

   Private Sub startbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles startbtn.Click
       Timer1.Enabled = True
   End Sub

   Private Sub stopbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
stopbtn.Click
       Timer1.Enabled = False
   End Sub

   Private Sub resetbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles resetbtn.Click
       t1 = 0
   End Sub
End Class
```

==
==

```
Public Class Form1
   Dim t1 As Decimal
   Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
       Select Case NumericUpDown1.Value
          Case 0
             t1 += 1
          Case 1
             t1 += 0.1
          Case 2
             t1 += 0.01
          Case 3
             t1 += 0.001
       End Select
       screenlbl.Text = Str(t1)
   End Sub

   Private Sub startbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles startbtn.Click
       Timer1.Enabled = True
   End Sub

   Private Sub stopbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
stopbtn.Click
```

```
        Timer1.Enabled = False
    End Sub

    Private Sub resetbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles resetbtn.Click
        t1 = 0
    End Sub

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NumericUpDown1.ValueChanged
        Select Case NumericUpDown1.Value
            Case 0
                Timer1.Interval = 1000
                t1 = Fix(t1)
            Case 1
                Timer1.Interval = 100
                t1 = Fix(t1 * 10) / 10
            Case 2
                Timer1.Interval = 10
                t1 = Fix(t1 * 100) / 100
            Case 3
                Timer1.Interval = 1
                t1 = Fix(t1 * 1000) / 1000
            Case Else
                NumericUpDown1.Value = 3
                Timer1.Interval = 1
                t1 = Fix(t1 * 1000) / 1000
        End Select
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        NumericUpDown1.Value = 0
        Timer1.Interval = 1000
    End Sub
End Class

==============================================================================
==

Public Class Form1
    Dim t1 As Decimal
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
        Select Case NumericUpDown1.Value
            Case 0
                t1 += 1
            Case 1
                t1 += 0.1
            Case 2
                t1 += 0.01
            Case 3
                t1 += 0.001
        End Select
```

```
        screenlbl.Text = Str(t1)
    End Sub
    Private Sub startbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles startbtn.Click
        Timer1.Enabled = True
        screenlst.Items.Add("start ad" & Str(t1))
    End Sub

    Private Sub stopbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
stopbtn.Click
        Timer1.Enabled = False
        screenlst.Items.Add("stop ad" & Str(t1))
    End Sub

    Private Sub resetbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles resetbtn.Click
        t1 = 0
    End Sub

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NumericUpDown1.ValueChanged
        Select Case NumericUpDown1.Value
            Case 0
                Timer1.Interval = 1000
                t1 = Fix(t1)
            Case 1
                Timer1.Interval = 100
                t1 = Fix(t1 * 10) / 10
            Case 2
                Timer1.Interval = 10
                t1 = Fix(t1 * 100) / 100
            Case 3
                Timer1.Interval = 1
                t1 = Fix(t1 * 1000) / 1000
            Case Else
                NumericUpDown1.Value = 3
                Timer1.Interval = 1
                t1 = Fix(t1 * 1000) / 1000
        End Select
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        NumericUpDown1.Value = 0
        Timer1.Interval = 1000
    End Sub
End Class
```

الفيديو رقم : 8

استخدام الجداول لإدخا ل أوإخراج بيانات كثيرة

```
Public Class Form1
```

```
    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        Dim i As Integer
        For i = Val(startnumtxt.Text) To Val(lastnumtxt.Text)
            screenlst.Items.Add(Str(i) + " ^ 2 = " + Str(i ^ 2))
        Next
    End Sub
End Class
```

==
==

```
Public Class Form1

    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        DataGridView1.Visible = True
        DataGridView1.Columns.Add("a1", "الرقم")
        DataGridView1.Columns.Add("a2", "الرقم مربع")
        Dim i, n As Integer
        For i = Val(startnumtxt.Text) To Val(lastnumtxt.Text)
            n = DataGridView1.Rows.Add()
            DataGridView1.Rows.Item(n).cells(0).value = Str(i)
            DataGridView1.Rows.Item(n).cells(1).value = Str(i ^ 2)
        Next
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        DataGridView1.Visible = False
    End Sub
End Class
```

==
==

```
Public Class Form1

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged
        If NumericUpDown1.Value > 0 Then
            DataGridView1.Visible = True
        Else
            DataGridView1.Visible = False
        End If
        Dim i As Integer
        For i = 1 To DataGridView1.Columns.Count
            DataGridView1.Columns.Remove("a" + Str(i))
        Next
        For i = 1 To NumericUpDown1.Value
            DataGridView1.Columns.Add("a" + Str(i), "رقم العمود" +
Str(i))
        Next
```

```
    End Sub

    Private Sub Form1_Load1(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        DataGridView1.Visible = False
        DataGridView2.Visible = False
    End Sub

    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        If NumericUpDown1.Value > 0 Then
            DataGridView2.Visible = True
        Else
            DataGridView2.Visible = False
        End If
        Dim i As Integer
        For i = 1 To DataGridView2.Columns.Count
            DataGridView2.Columns.Remove("a" + Str(i))
        Next
        For i = 1 To NumericUpDown1.Value
            DataGridView2.Columns.Add("a" + Str(i), "رقم العمود" +
Str(i))
        Next
        Dim n As Integer = DataGridView1.Rows.Count - 1
        Dim sum As Integer
        For i = 1 To n
            sum += DataGridView1.Rows.Item(i - 1).Cells(0).Value
            DataGridView2.Rows.Item(0).Cells(0).Value = sum

        Next
    End Sub
End Class

=========================================================================
==

Public Class Form1

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged
        If NumericUpDown1.Value > 0 Then
            DataGridView1.Visible = True
        Else
            DataGridView1.Visible = False
        End If
        Dim i As Integer
        For i = 1 To DataGridView1.Columns.Count
            DataGridView1.Columns.Remove("a" + Str(i))
        Next
        For i = 1 To NumericUpDown1.Value
            DataGridView1.Columns.Add("a" + Str(i), "رقم العمود" +
Str(i))
        Next
    End Sub
```

```
    Private Sub Form1_Load1(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        DataGridView1.Visible = False
        DataGridView2.Visible = False
    End Sub

    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        If NumericUpDown1.Value > 0 Then
            DataGridView2.Visible = True
        Else
            DataGridView2.Visible = False
        End If
        Dim i, j As Integer
        For i = 1 To DataGridView2.Columns.Count
            DataGridView2.Columns.Remove("a" + Str(i))
        Next
        For i = 1 To NumericUpDown1.Value
            DataGridView2.Columns.Add("a" + Str(i), "رقم العمود" +
Str(i))
        Next
        Dim n As Integer = DataGridView1.Rows.Count - 1
        Dim sum(NumericUpDown1.Value) As Integer
        Dim average(NumericUpDown1.Value) As Decimal
        For j = 1 To NumericUpDown1.Value
            For i = 1 To n
                sum(j) += DataGridView1.Rows.Item(i - 1).Cells(j -
1).Value
            Next
            DataGridView2.Rows.Item(0).Cells(j - 1).Value = sum(j)
            average(j) = sum(j) / n
        Next
        i = DataGridView2.Rows.Add
        For j = 1 To NumericUpDown1.Value
            DataGridView2.Rows.Item(i).Cells(j - 1).Value = average(j)
        Next
    End Sub
End Class

==================================================================
==
Public Class Form1

    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        If usernametxt.Text = "brahim" And passwordtxt.Text = "aklim-
maroc" Then
            MsgBox("لقد نجحت في الدخول")
        Else
            MsgBox("تأكد من بيانات الدخول")

        End If
    End Sub
End Class
```

هذا الكود ليس له برنامج مسجل كباقي الأكواد، للإستئناس فقط.

===
=

برنامج متعدد الشاشات (3 فورمات)

```
Public Class Form1

    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        If usernametxt.Text = "brahim" And passwordtxt.Text = "aklim-
maroc" Then
            Form2.Show()
            Me.Hide()

        Else
            MsgBox("تأكد من بيانات الدخول")

        End If
    End Sub
End Class

Public Class Form2

    Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles resultbtn.Click
        Form3.Show()
        Me.Hide()
    End Sub
End Class

Public Class Form3

    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        DataGridView1.Visible = True
        DataGridView1.Columns.Add("ًa1", "الرقم")
        DataGridView1.Columns.Add("a2", "مربع الرقم")
        Dim i, n As Integer
        For i = Val(Form2.startnumtxt.Text) To
Val(Form2.lastnumtxt.Text)
            n = DataGridView1.Rows.Add()
            DataGridView1.Rows.Item(n).Cells(0).Value = Str(i)
            DataGridView1.Rows.Item(n).Cells(1).Value = Str(i ^ 2)
        Next
    End Sub

    Private Sub exitbtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles exitbtn.Click
        Application.Exit()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
        Form2.Show()


    End Sub
End Class
```

هذا الكود نسخته من البرنامج قبل إدخال الفورم الرابع النهائي

======================================

```
Public Class Form1

   Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles resultbtn.Click
      If usernametxt.Text = "brahim" And passwordtxt.Text = "aklim-maroc" Then
         Menuform.Show()
         Me.Hide()

      Else
         MsgBox("الدخول بيانات من تأكد")

      End If
   End Sub
End Class

Public Class Form2

   Private Sub resultbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles resultbtn.Click
      Me.Hide()
   End Sub
End Class

Public Class Form3

   Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
      DataGridView1.Visible = True
      DataGridView1.Columns.Add("ًa1", "الرقم")
      DataGridView1.Columns.Add("a2", "الرقم مربع")
      Dim i, n As Integer
      For i = Val(Form2.startnumtxt.Text) To Val(Form2.lastnumtxt.Text)
         n = DataGridView1.Rows.Add()
         DataGridView1.Rows.Item(n).Cells(0).Value = Str(i)
         DataGridView1.Rows.Item(n).Cells(1).Value = Str(i ^ 2)
      Next
   End Sub

   Private Sub exitbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
exitbtn.Click

      Me.Hide()
   End Sub
End Class
```

```
Public Class Menuform

   Private Sub ToolStripMenuItem6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripMenuItem6.Click
       Application.Exit()
   End Sub

   Private Sub MainDataToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MainDataToolStripMenuItem.Click
       Form2.Show()
   End Sub

   Private Sub ResultToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ResultToolStripMenuItem.Click
       Form3.Show()
   End Sub
End Class
```

انتهى الكتاب بعون من الله العلي القدير ، وأسأل الله أن ينتفع به الجميع